
INPUT
Publicación práctica para usuarios de
commodore
Septiembre 1985 Precio 350 Ptas
Año 1 Número 1
Contiene un coleccionable
VISTA Y OIDO PARA EL C-64
EL JUEGO DE LA VIDA
ENVIA MENSAJES SECRETOS
SECCION DE SOFTWARE
VOICE MASTER
LEVEL
COVOX INCORPORATED
patent pending


STARS
COMMODORE
El SOFTWARE a la altura
de tu ordenador.
STARS
COMMODORE
CADA MES EN TU KIOSKO



DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Christophe Pais, Francisco Tórtola, José Pérez Montero, Benito Román

INPUT Commodore es una publicación juvenil de EDICIONES FORUM
GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ Cuenca, 1, 1.º
Tel. 253 45 01 /02
Barcelona: Avda. de Sarriá, 11-13, 1.º
Tel. 250 23 99
FOTOMECANICA: Ochoa, S. A.
COMPOSICION: Fernández Ciudad, S. L.
IMPRESION: Edime, S. A.
Depósito legal: M. 27.884-1985

Suscripciones: EDISA,
López de Hoyos, 141. 28002 Madrid
Tel. (91) 415 97 12
Redacción:
Alberto Alcocer, 46, 4.º
28016 Madrid. Tel. 250 10 00
DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona

Se solicitará el control OJD

INPUT Commodore es independiente y no está vinculada a Commodore Business Machines o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

INPUT commodore

Sumario

AÑO 1 NUMERO 1

YA ESTAMOS AQUI

¡Una revista más!, será la reflexión que haréis muchos lctores. Efectivamente así es, salimos a la calle para ampliar la oferta disponible en los kioscos. Sin embargo, nacemos con pretensiones, resumidas en un objetivo único: ser el centro de vuestras preferencias. Intentaremos lograrlo número a número.

Hemos creado diversas secciones que suponemos serán del interés de la mayoría de vosotros. Conviene hacer especial mención de la titulada Educación. En los primeros números se describe el **Logo** y su modo de empleo, pero trataremos otros temas relacionados con dicha temática.

En las páginas centrales se incluye una sección que enseña a programar juegos. El grado de complejidad irá aumentando, tratándose diversos tipos, desde los más sencillos a los más complejos (juegos de guerra e inteligencia), partiendo desde el comienzo de lo más básico. Esta sección puede ser coleccionada por quienes así lo decidáis.

El *software* alcanza una especial relevancia en **Revista de Software,** sección en la que, mes tras mes, haremos repaso de las novedades más interesantes. Los programas se describen con detalle, comentando diversos trucos y tácticas. Cada programa tiene su ficha técica y una calificación de sus diversos aspectos. Y, cómo no, se incluyen numerosas fotografías de pantallas.

A lo largo de la revista podréis encontrar diversos concursos. Desde luego, no perseguimos captar a coleccionistas de regalos, pero sí corresponder a vuestra participación.

Confiamos en que **INPUT** sea de vuestro agrado, pero os invitamos a que nos escribáis con vuestras sugerencias.

MEJORAS Y APLICACIONES

Cualquier programa publicado en INPUT puede sugerir nuevas ideas. Te lanzamos una propuesta a modo de reto: mejora y/o encuentra aplicaciones a nuestros listados. No olvides que muchas veces salta una chispa que trae una solución genial.

Cada tercer mes, a partir de la publicación de un número, elegiremos la mejor aportación.

La decisión será inapelable, pero tener por seguro que será ecuánime y, sobre todo, resuelta por un grupo de expertos.

El ganador recibirá un premio consistente en una suscripción por un año a INPUT COMMODORE.

EL BUZON DE INPUT

El próximo número de INPUT reservará un espacio para responder a vuestras cartas.

No pretendemos que sea un rincón únicamente destinado a la resolución de dudas y problemas. Por supuesto que intentaremos proporcionaros este tipo de ayuda, pero nuestra pretensión es tener algo más: un espacio para la comunicación donde nos contéis a todos lo que hacéis con el ordenador, proyectos, intenciones, cómo es vuestra instalación y las cien mil cosas que se os ocurran.

Dirigir las cartas a:

EL BUZON DE INPUT
COMMODORE
Alberto Alcocer, 46, 4.º B
Madrid

:RITEMAN:

PERSONAL/BUSINESS PRINTER

AMPLIA GAMA

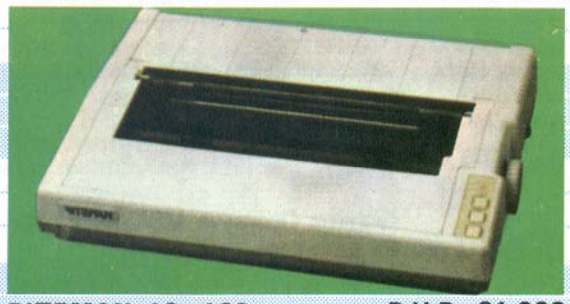
Nuevas impresoras modelos **F+** y **C+**, sin rodillo alimentación horizontal, impresión vertical, tracción y fricción desde 4 a 10", bidireccional optimizada velocidad 105 cps. con soportes de elevación.



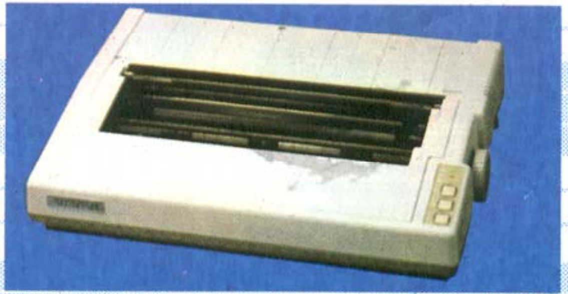
RITEMAN F+: Interface Paralelo Centronics, 2K buffer NLQ — **P.V.P. 69.000 pts.**

RITEMAN C+: Especial directa a COMMODORE (cable inc.) — **P.V.P. 67.000 pts.**

Otros modelos RITEMAN en 80 y 136 columnas, velocidad 120, 140, 160 cps.



RITEMAN 10, 120 cps. P.V.P. 81.000



RITEMAN 10-IBM, 140 cps. P.V.P. 85.000



RITEMAN 10-II 160 cps. P.V.P. 93.000



RITEMAN 15 160 cps. P.V.P. 155.000

DE VENTA EN LOS MEJORES ESTABLECIMIENTOS ESPECIALIZADOS

DATAMON

DATAMON, S. A.

PROVENZA, 385-387, 6.º, 1.ª
TELÉFONO (93) 207 27 04

REPRESENTACION EN ESPAÑA DE:

:RITEMAN:

-IMPRESORAS PROFESIONALES-

08025 - BARCELONA

* **MAYORES PRESTACIONES**
* **MENOR TAMAÑO**
* **MEJOR PRECIO**



¿JACK TRAMIEL EN ESPAÑA?

Se dice, se comenta, se afirma que Jack Tramiel, fundador y expresidente de Commodore, visitó España durante los primeros días del pasado mes de julio.
Tramiel se situó a la cabeza de Atari, después de los problemas que le condujeron a dejar la presidencia de Commodore en los primeros meses del 85.
Al parecer su visita tendría como objetivo principal entrevistarse con un importante distribuidor nacional. Las especulaciones van más lejos de suponer que se pretende una implantación más fuerte de Atari en nuestro país, incluso llega a comentarse la posibilidad de fabricación, aun cuando Tramiel tiene resuelto el problema en Taiwan, donde se encuentra mano de obra a bajo precio.
Los nuevos modelos de Atari compiten en los mismos mercados que Commodore y Sinclair.
Unos días antes, a mediados de junio, Tramiel llegó a afirmar en Francia que se hallaba en conversaciones con el gobierno de aquel país de cara a participar en un plan de fabricación de cien mil ordenadores en colaboración con un fabricante francés, de quien no especificó el nombre.
Igualmente expresó su idea de crear dos nuevas fábricas, en Europa y los EE.UU.
Habrá que esperar acontecimientos. Tal vez el cercano SIMO traiga noticias interesantes para la Distribución.



WALT DISNEY PREPARA SOFTWARE

Walt Disney Productions se ha dejado seducir por el suculento mercado que los microordenadores le brindan a sus populares personajes. En los EE.UU. ya ha lanzado tres títulos en *diskette* : Mickey's Space Adventure, que dispone de 150 diferentes pantallas visualizables en las que el famoso ratón protagoniza una aventura educativa por el espacio, explicando a los mas pequeños los conocimientos básicos del Universo.
Winnie The Pooh In The Hundred Acre Wood adentra a los chiquitines en la lectura y el razonamiento lógico.
El tercer paquete responde a Donald Duck's Playground. Aquí el siempre malhumorado pato enseña como progresar en los negocios y ganar dinero empleando la cabeza y la lógica.
Para agosto de este año se ha previsto la presentación de una nueva versión del Mago de Oz, pelicula que llevará por título Return of Oz (La vuelta de Oz) y no tardará en estar lista una versión del programa en base a la pelicula.
El libro de la selva es otra cinta que dará proximamente paso a nuevo *software*.



SUCESOR PARA EL C-64

En el momento en que estas líneas alcancen el kiosco es posible que el nuevo Commodore 128 esté en los escaparates o a punto de aparecer, por un precio que próximo a las 80.000 pts.
El C-128 es el descendiente natural del C-64 e incorpora suculentas mejoras con respecto a él.
Tal vez la parte más compleja y delicada del lanzamiento de un nuevo ordenador consiste en convencer a las casas de *software* para que desarrollen programas que potencien su manejo por parte del consumidor. Un magnífico ordenador sin *software* de verdadera calidad que lo arrope no tiene las mismas posibilidades de salir haciendo ruido que otro capaz de aprovechar una biblioteca de programas previamente existente. Este último es el caso del C-128, que puede funcionar en tres modos diferentes: C-64, C-128 y CP/M.
En el primer modo de funcionamiento es realmente un 64 lo que hay bajo sus teclas, incluso dispone de un microprocesador compatible con el 6510. De esta manera se evitan complicaciones como las sufridas por los modelos Plus 4 y C-16, que no son compatibles con otros modelos de la firma (incluso los conectores del *cassette* y los *joysticks* son diferentes). El primero de ellos ni siquiera se ha llegado a comercializar aquí.
El modo C-128 emplea el microprocesador 8502, compatible con el 6510 y capaz de direccionar hasta 128 Kbytes de memoria. Posee un monitor de lenguaje máquina incluido y conserva los *Sprites*, aparte de poder trabajar también con formato de 80 columnas en la pantalla. Asimismo emplea una versión más avanzada del BASIC, bautizada como 7.0 (la del 64 es la 2.0).
El modo CP/M también existía para el C-64, pero en forma de cartucho enchufable exteriormente y complementado con un *diskette*. En modo C-128 trabaja con 80 columnas de texto y tiene incorporado el microprocesador Z80 para trabajar con el sistema operativo CP/M



EL ORDENADOR DE PULSERA



La firma japonesa Seiko ha lanzado recientemente una replica de lo que podría ser el primer ordenador de pulsera. El RC-100 es un terminal multifunción de cuarzo, que se puede conectar al *port* del usuario de un Commodore 64. Esta conexión posibilita que la información sea volcada en el "reloj" y puedan pedirse los datos de forma inmediata.
Tiene cinco modos de trabajo: hora, área de almacenamiento definible por el usuario, horario mundial, alarma semanal y alarma planificada, que permite memorizar hasta ochenta eventos anuales, tales como entrevistas. Análogamente, es capaz de crear doce ficheros separados, que almacenarán el listín telefónico, referencias de precio, etc.



SOFTWARE PARA CONTROLAR EL CUERPO

La firma británica Mirrorsoft es la filial productora de *software* del grupo editorial propietario del diario Daily Mirror.
En los últimos tiempos está obteniendo interesantes críticas en el mercado británico, principalmente por dos programas destinados al consumo humano: Know Your Own Psi-Q (conoce tus poderes síquicos) y The Joffe Plan.
El primero de ellos es un conjunto de programas que sirven para testear los poderes de percepción extrasensorial de quien lo utiliza. Un cuestionario permite centrar al individuo en la cuestión, proporcionandole las indicaciones pertinentes. La continuación incluye cuatro juegos interactivos para comprobar la clarividencia y similares.
El segundo es un plan basado en un análisis de las razones que hacen que las dietas estén condenadas al fracaso. Se personalizan los datos por el conocimiento de los hábitos alimenticios y forma de vida, tomando en cuenta todos los factores que afectan al peso.



UNIDADES DE ALMACENAMIENTO ALTERNATIVAS



Los elevados precios de las unidades de *diskette* hacen que los usuarios de ordenadores personales busquen unidades alternativas de almacenamiento masivo. El *cassette* no es lo suficientemente rápido ni fiable como para agradar completamente al usuario. Situado entre ambos dispositivos aparece uno que utiliza *Wafers* (tabletas), que son cartuchos de cinta magnética, similares los empleados por el Microdrive de Sinclair. Poseen una velocidad intermedia y mayor fiabilidad.
El modelo que mayor relevancia está tomando es el Quick Data Drive, de la Entrepro. Dispone de sus propios controlador y sistema operativo.



CADA VEZ MAS RAPIDO



La unidad 1571 de Commodore se caracteriza por su mayor velocidad en la transferencia de datos y el perfíl estrecho de la carcasa. Puede trabajar con todos los microordenadores domésticos de la firma y es compatible con las unidades 1541 y 2031.
Está concebido para complementar al nuevo C-128 y por eso además es compatible con el sistema operativo CP/M, pudiendo grabar y leer datos en los formatos empleado por otros conocidos ordenadores, pudiendo utilizar todo su *software*.



COMMODORE FILIAL EN FRANCIA

La firma que venía distribuyendo Commodore en el vecino pais ha dado paso a la llegada de una nueva filial europea de la multinacional.
Es tradición que las firmas relacionadas con la informática decidan instalarse en los paises cuyo mercado alcanza determinadas cotas en el desarrollo en las ventas, pasando normalmente por absorber al anterior distribuidor oficial.

COPIAR, COPIAR HASTA MORIR

Adapso, una asociación americana que aglutina a los mas importantes fabricantes de *software*, ha lanzado un agresiva campaña publicitaria destinada a crear conciencia entre quienes realizan copias ilegalmente. Comienza uno de sus anuncios con cuatro lineas de texto que dicen:

"Es fácil hacer una copia.
Es rápido.
Es ilegal.
Es incorrecto."

Continua haciendo una reflexión no por evidente menos cierta: *"Es difícil de creer. Gente que no habría pensado en sustraer un producto de software de una tienda durante su hora de comida, no se lo piensa dos veces a la hora de regresar a la oficina y hacer varias copias ilegales de un mismo software."*

Termina ofreciendo un panfleto, que gustosamente enviará la srta. Priscila a quienes se tomen la molestia de escribir a sus oficinas.

El problema tiene visos de gravedad, al menos en el terreno económico. Una encuesta llevada a cabo a principios del presente año revelaba que a la industria británica del *software* la pirateria le cuesta venticuatro millones de Libras esterlinas anuales. Casi nada!. Quienes parecen llevar la peor parte son los fabricantes de programas de juegos, que se sienten mas vulnerables que las firmas que centran su actividad en el software de gestión.

El informe alcanza tales grados de sutileza, que incluso se establecen categorias en la especialidad: Falsificaciones, imitaciones que parten de una idea básica ajena, copias hechas por minoristas para favorecer la venta de un equipo, copieteo aislado entre amiguetes, la copia mas o menos organizada que llevan a cabo los estudiantes y la rotación del personal que trabaja en las empresas del sector.

EL GUARDADO SECRETO DE UN AMIGA

Según el actual presidente de Commodore Business Machines, Marshall Smith, el moderno ordenador Amiga podría estár circulando por Europa a principios del próximo año. Esto muy bien podría significar que los distribuidores europeos de la marca tengan al menos un prototipo con el que ir familiarizandose.

El diseño es una maravilla. Utiliza como procesador central un miembro de la controvertida familia Motorola 68000, que comparte caracteristicas comunes a los microprocesdores de 32 y 16 bits. Es el mismo que emplea el fabuloso Mackintosh de la firma Apple Computer y el QL de Sinclair (en versión simplificada).

Sin embargo, Commodore lo mantiene oculto, salvo la presentación que hizo a la prensa y sus distribuidores en la Feria de la Electronica de Consumo, celebrada en Chicago. No quiere que se lo "birle" la competencia y la mejor forma de conseguirlo y no dar oportunidades para que lo copien. Mientras tanto son varias las casas de software que disponen de una unidad para desarrollar las aplicaciones que catapultarán al ordenador Amiga.

Alberga en su configuración básica a 192 Kbytes de ROM, 256 Kbytes de RAM y una unidad para *diskettes* de 3,5 pulgadas, que trabaja en doble cara y es capaz de almacenar hasta 800 K por disco.

Se puede conseguir una visualización sensacional a partir de la salida para monitor en color RGB (los colores rojo, verde y azul van por separado).

El Amiga se apunta a la actual tendencia a emplear *ventanas* multiples en la pantalla, soportadas por el sistema operativo. Este último recibe el curioso y sugestivo nombre de Intuición.

Lo cierto es que todos los que lo han visto coinciden en afirmar que es una máquina con cualidades excepcionales. Incorpora tres *chips*, especialmente diseñados para ocuparse del sonido, los gráficos y animación (fabulosa, casi dibujos animados) y la gestión de los periféricos (en los C-64 y C-128 las funciones de visualización y sonido son también son acometidas por *chips* especificos).

La gama de colores manejable por el Amiga tiene 4.096 posibilidades. Dispone de un *interface* que le permite mezclar las imagenes de produccion propia con otras externas, procedentes de cualquier fuente de video,tal como una camara, un videoreproductor, etc.. El sonido tambien puede ser escuchado en estereo.

Un cartucho le convierte en compatible con el PC de IBM, poniendo a su alcance el sistema operativo MS-DOS. El cartucho incorpora al microprocesador 8086, el mismo empleado por el modelo de IBM.

Su lanzamiento disipará las expectativas, pero el impacto parece asegurado.

PROXIMAS NOVEDADES

El Commodore Show es el principal evento commodorero en Europa y en el Commodore Business suele presentar algunas novedades.

La última edición del certamen sirvió para tener la certeza de que el C-64 seguirá siendo un ordenador plenamente vigente, pues lo presentado asi lo indicaba.

Dos productos de Commodore a destacar: el Sampler (muestreador) y un sensacional tenis. El primero es una versión ampliada de un cartucho analizador de sonidos, capaz digitalizarlo a través de un microfono y hacer con el mil diabluras, gracias al *software* que lo complementa. Se prevé que esté disponible en Navidad.

El tenis no es algo novedoso, pero esta versión es algo muy sofisticado.

ASI LO OYE TU ORDENADOR

- MUNDOS ANALOGICO Y DIGITAL
- LECTURA DE UNA SEÑAL
- LA TRAZA SONORA
- VISUALIZACION

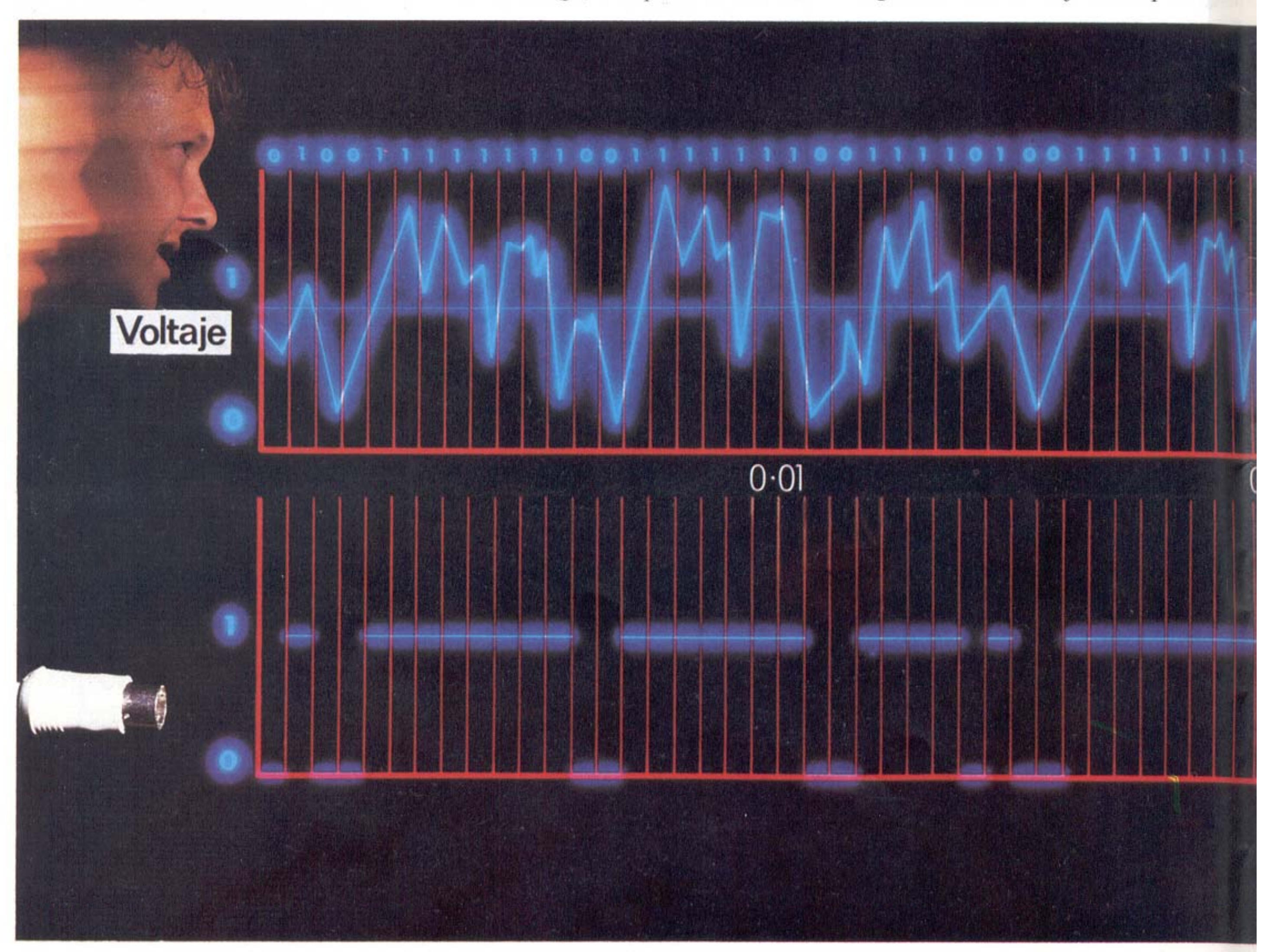
Ya puedes utilizar tu ordenador doméstico para introducirte en la tecnología de la digitalización del sonido. Un sencillo programa te permitirá analizar sonidos con tu microordenador.

Todo sonido tiene dos componentes —frecuencia y volumen—, siendo capaz el oído humano de analizar e interpretar estas señales complejas que son, literalmente, vibraciones del aire que le rodea, admitiéndolas como sonidos reconocibles.

Visto de tal forma, el sonido es una señal analógica, es decir, que varía de una forma continua dentro de un amplio margen, siendo cada variación significativa. A diferencia de los sentidos humanos, los ordenadores no son capaces de reconocer esta clase de cambios, necesitando por el contrario una señal digital. En esta señal cada variación viene representada por un valor claramente distinto, un 0 o un 1, una presencia o una ausencia de señal.

Sin embargo, aunque el ordenador no puede interpretar directamente una señal sonora, es muy sencillo convertir la forma de onda analógica, por ejemplo de un fragmento musical, en información digital lista para ser utilizada.

En las aplicaciones prácticas esta técnica se sitúa al frente de la tecnología de grabación del sonido, que está evolucionando desde el tradicional almacenamiento en cinta (de la señal analógica) a sistemas computerizados con almacenamiento en disco magnético. Las ventajas son que una



vez almacenado el sonido en forma digital es mucho más fácil modificarlo o combinarlo con otros sonidos, y una vez grabado es mucho menor el riesgo de «corrupción» debido a las limitaciones impuestas por el sistema de grabación.

MUSICA EN TU MICRO

Aunque actualmente esta tecnología solamente existe en unos cuantos sistemas caros y altamente sofisticados, un ordenador doméstico es capaz de ofrecer una aproximación a estas posibilidades.

Cada vez que cargas un programa (con LOAD) almacenado en una cinta de *cassette,* se está reproduciendo una señal que había sido almacenada digitalmente. Y como sabrás, si has escuchado alguna vez por el altavoz una cinta de ordenador, la señal produce además un tipo de sonidos que, aunque digitales, no admiten una interpretación con significado coherente al oído.

Pero con una adecuada programación (que comporta la utilización del código máquina), puedes utilizar tu sistema para poner música en el micro. Lo único que debes hacer es introducir en la máquina una señal analógica por el conector del *cassette,* y decir al ordenador cómo interpretarla, convirtiéndola en una señal digital.

Una vez que el ordenador ha hecho esto, puedes almacenar los números obtenidos en su memoria, e incluso visualizarlos en pantalla, como muestra el siguiente programa analizador de sonidos. Esto permite que un ordenador doméstico convierta la señal analógica de cualquier sonido grabado en una serie de números, que pueden utilizarse posteriormente para producir una traza gráfica en la pantalla, o ser almacenados en memoria, para la reproducción en otro momento, e incluso dentro de otro programa de ese sonido previamente digitalizado.

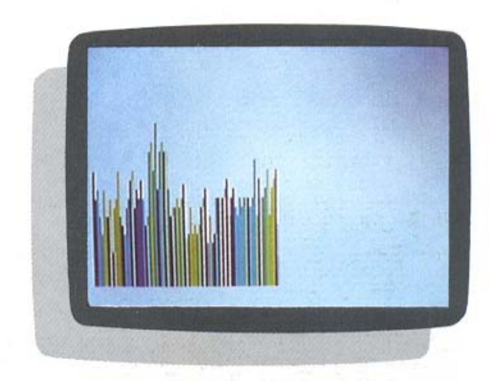
TRAZA SONORA

El programa que te ofrecemos a continuación sirve para producir una representación gráfica, continuamente cambiante, de la música o sonido reproducidos en el magnetófono conectado al micro. Cuando se pone en marcha la cinta aparece en la pantalla una serie de líneas muy próximas cuya longitud depende directamente del sonido que se esté analizando: cuanto más alta es la frecuencia, más larga es la línea. Cuando la pantalla se llene, la traza desaparecerá y volverá a empezar a partir del lado izquierdo.

Una señal analógica que entra al port *del* cassette *se muestrea 2000 veces por segundo. Cuando su valor está por encima de la tensión umbral se detecta y se graba un 1. Las señales por debajo de este nivel dan un 0. La traza producida digitalmente de esta manera imita, en efecto, la forma de onda analógica del sonido.*



Estas pantallas muestran las trazas realizadas por el ordenador a partir de dos melodías.



COMO FUNCIONA

Como el ordenador no puede interpretar directamente las variaciones de una señal analógica de sonido, se le programa para ir asignando a la misma valores digitales. Lo que realmente hace es tomar muestras de la señal en el *port* del *cassette* a intervalos muy rápidos, miles de veces por segundo. La señal en el *port* del *cassette* sólo puede ser alta o baja, representada por un 1 o un 0; no hay valores intermedios como en una señal analógica. Pero como la velocidad de muestreo es tan rápida, la velocidad de cambio de la señal digital imita la forma de onda de la señal analógica.

Imagina, por ejemplo, que se introduce una señal con una frecuencia de 256 Hz (es la nota Do central), en cuyo caso la señal alcanzará 256

máximos en un segundo y cada máximo permanecerá durante 1/512 de segundo. Si se realizan lecturas en el *port* a razón de 2000 lecturas por segundo, entonces cada máximo resultará leído durante aproximadamente cuatro tomas de muestras. A continuación la señal leerá un mínimo en otras cuatro muestras y así sucesivamente. De esta forma las variaciones de los valores digitales siguen la forma de onda de una manera más o menos precisa. Cuanto más rápidamente se realice este muestreo, mucho mejor captará el ordenador los cambios sutiles de la señal analógica.

El programa utiliza los valores digitales que obtiene de las maneras descritas anteriormente. Para representar gráficamente la forma de onda, dibuja una figura mostrando cuántos bits altos encuentra por unidad de tiempo, es decir, la frecuencia global del fragmento de música de que se trate. Aquí también, cuanto más frecuente sea el muestreo y más frecuente el dibujo, más precisa será la representación gráfica.

Los datos en código máquina para la subrutina que lee el *port* del *cassette* están contenidos dentro del programa principal como DATAs, que comienzan en la línea 1000. Una vez introducido el programa, hágase un SAVE a cinta o disco antes de ejecutarlo con RUN. Después ejecuta el programa. Como la subrutina es muy corta, los datos en código máquina se almacenan en el *buffer* del *cassette*. Cuando se ha ejecutado el programa, aparece en pantalla el mensaje INSERTA LA CINTA Y PRESIONA PLAY. DESPUES PULSA RETURN.

Si deseas congelar la traza sonora en algún punto particular, pulsa cualquier tecla. La imagen de la pantalla quedará inmóvil hasta que sueltes la tecla apretada.

Nota: Este programa no funciona en aquellas unidades que no dispongan del cartucho de ampliación **Simon's BASIC** dado que en él se utilizan comandos de BASIC ampliado.

```
   1 REM*ES NECESARIO EL
     SIMON'S BASIC
   5 FOR Z=832 TO 852:READ
     X:POKE Z,X:NEXT Z
   8 PRINT"♥INSERTA LA
     CINTA Y PRESIONA PLAY"
   9 PRINT"DESPUES PULSA
     RETURN":INPUT A$
  1Ø HIRES Ø,1:COLOUR
     1,1:MULTI 3,4,7
  2Ø FOR Z=Ø TO 159
  3Ø SYS 832
  4Ø IF PEEK(251)*4<2ØØ THEN
     LINE Z,2ØØ,Z,2ØØ-
     PEEK(251)*4,RND(1)*3+1:
     GOTO 5Ø
  45 IF PEEK(197)<>64 THEN 45
  5Ø NEXT Z:GOTO 1Ø
1ØØØ DATA 162,Ø,134,251,173,
     13,22Ø,197,252,24Ø,6,133,
     252,23Ø,251,24Ø,3,232
1Ø1Ø DATA 2Ø8,24Ø,96
```



COMO UN LORO: ESCUCHA Y REPITE

- ALMACENAR LA VOZ
- DIGITALIZAR LA SEÑAL
- EN MARCHA
- UTILIZAR EL CHIP
- EL SOFT ES IMPORTANTE

Si decimos que un ordenador es capaz de producir voz comprensible, a nadie sonará como novedad. Pero si dicha voz es la nuestra, el asunto comienza a ser interesante. El Voice Master **es un conjunto** *hard-soft* **que permite que introduzcamos nuestro habla en las «neuronas» de un** Commodore 64.

Desde antiguo el hombre soñó con la posibilidad de crear todo tipo de máquinas capaces de emular el comportamiento humano y, por extensión, el de la naturaleza. Pero no fue hasta el advenimiento de las recientes tecnologías relacionadas con la informática cuando empezó a vislumbrar tales posibilidades.

En los primeros años posteriores al desarrollo de los ordenadores, un audaz programador dio brincos de alegría cuando logró hacer que una impresora lanzase notas musicales mediante su mecanismo de impresión, controlado por el ordenador. Desde entonces se ha investigado con avidez, obteniendo asombrosos resultados.

La revolución también ha llegado a los modestos. En los últimos tiempos han surgido diversos dispositivos capaces de arrancarle unas palabras al ordenador personal. Sin embargo, las técnicas empleadas son diversas.



Hardware *del* **Voice Master:** *un módulo digitalizador y un micrófono.*

ALMACENAR LA VOZ

La voz dispone de una serie de características muy particulares, que deben ser almacenadas si pretendemos reproducirlas con posterioridad. No obstante, es factible reducirlas a una serie de números rápidamente cambiantes en el dominio del tiempo. Almacenándolos podemos controlar un determinado tipo de dispositivo electrónico, que nos proporcione el resultado apetecido. El problema principal consiste en las grandes cantidades de capacidad de memoria que se precisan para guardar tanto número y que, además, deben estar disponibles tan de inmediato, que si las guardamos en una unidad de almacenamiento masivo, por ejemplo una de *diskettes,* no conseguimos el propósito. Por tanto, la información ha de permanecer en la memoria central (RAM en este caso). Aunque los datos sí podrían guardarse en el *diskette* mientras no deban ser utilizados de inmediato.

La necesidad es la madre de la invención; por lo tanto, no es de extrañar que a este problema se le hayan dado mil vueltas.

Indudablemente, el método más racional para abordar la solución a la síntesis de voz parece recaer en la imitación de los órganos productores

LOS COMANDOS DEL VOICEMASTER

El *software* permite almacenar hasta 63 palabras o frases denotadas aquí por la variable N.

LEARN N	Registra la frase o palabra N.
SPEAK N	Reproduce la frase o palabra N.
SPEED J	Velocidad de reproducción (J entre 0 y 9).
RATE J	Altera la frecuencia del muestreo
VOLUME J	Altera el volumen de reproducción (0 a 15).
PUT "fichero", 8 ó 1	Guarda el vocabulario en cinta o *diskette.*
FIND "fichero", 8 ó 1	Carga el vocabulario desde cinta o *diskette.*
CLEAR	Borra el vocabulario y restituye los valores iniciales.
SCREEN	Desactiva la pantalla durante la reproducción.

Estos comandos se pueden combinar normalmente en un programa BASIC sin tomar precauciones especiales, solamente cargar VM o VM2. Se pueden guardar y volver a reclamar cuantos juegos de vocabulario se deseen, tanto en cinta como en *diskette,* con la ayuda de PUT y FIND. Cada uno de ellos podrá constar, como máximo, de 63 frases o palabras.

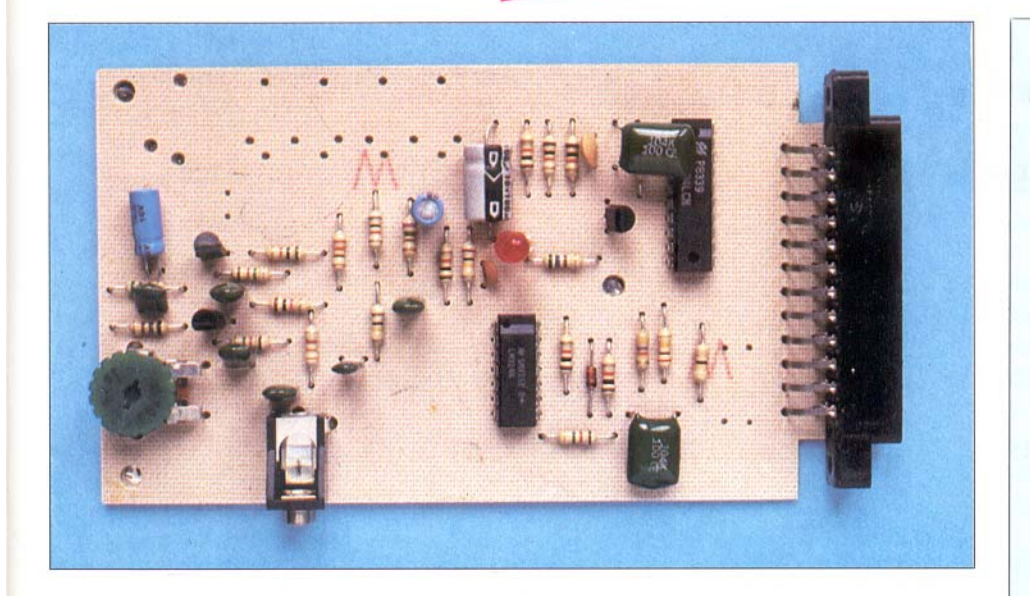
del habla por parte del ordenador, emulando el funcionamiento de los pulmones, las cuerdas vocales, laringe, cavidad bucal, etc. Esto es factible realizarlo con grandes ordenadores capaces de procesar millones de datos por segundo. Pero este tipo de síntesis predictiva requiere ingentes cantidades de memoria. El resultado es bueno, pero demasiado perfecto, y produce una voz similar a la de los robots de las películas —sería necesario introducirle las imperfecciones propias de los órganos humanos.

El método más ampliamente adoptado para los microordenadores consiste fundamentalmente en un ahorro de memoria. Se toman *alófonos* o *fonemas* y se almacenan en una memoria ROM (sólo pueden leerse los datos dispuestos en ella por el fabricante). Posteriormente se reproducen según lo requiera el programa que ejecuta el ordenador. El resultado es aceptable y su funcionamiento es similar a lo que podríamos hacer con un magnétofono si grabásemos una serie de fonemas fundamentales y uniéramos varios de ellos, pegando los trozos de cinta adecuados entre sí, para obtener una palabra. El *chip* más ampliamente utilizado para este menester en los microordenadores es el **SPO256,** utilizado por el **Currah Speech,** que contiene 64 de estos alófonos en su interior, que pueden ser combinados entre sí para formar palabras y frases. De esta forma se obtiene una importante economía de memoria, porque las componentes fundamentales se encuentran en la ROM del *chip,* que nada tiene que ver con la memoria propia del ordenador. El principal problema que surge es la procedencia americana del *chip,* que se traduce en que los alófonos han sido seleccionados entre los que se utilizan habitualmente en el inglés hablado. Cuando se componen palabras en castellano, la voz se parece al doblaje de las películas de **El Gordo y el Flaco.** Otra solución utilizada es simplemente una ampliación de la anterior, en el módulo que se conecta al ordenador se ha incluido una ROM que almacena un conjunto limitado de palabras completas, que deberemos combinar para obtener frases habladas. El sonido resulta más natural, porque se parece mucho a una grabación en cinta.

Y llegamos al protagonista de nuestro artículo, el **Voice Master.** Esta solución tiene algo que ver con la anterior, pero en este caso nos libramos de la tiranía impuesta por la ROM. Somos nosotros quienes podemos grabar la propia voz, pero no lo hacemos en una ROM, sino que la depositamos directamente en un área especial de la RAM que el *software* proporcionado por el fabricante asigna.

DIGITALIZAR LA SEÑAL

El **Voice Master** se compone, como decíamos antes, de dos partes: *hardware* y *software.* La primera consiste en un módulo digitalizador, que convierte el sonido en datos numéricos primarios, que el ordenador procesará. La fuente original puede ser el micrófono, tipo **electret** (con una amplia y plana respuesta en frecuencias, para quienes interese el dato) u otra fuente, tal como un *cassette* de audio. El módulo es una caja rectangular de aluminio, que se conecta en la parte posterior del **C-64** al *port* del usuario. En la parte superior dispone de un diodo luminoso (LED) que indica si el nivel del sonido es adecuado; si no lo fuera, es posible efectuar un ajuste con un destornillador que podemos introducir en otro agujero que existe en la caja.



Aspecto interior del módulo del **Voice Master.**

A QUIENES PUEDA INTERESAR

El módulo *hardware* que se conecta no es ninguna suerte de pieza mágica. En su interior se encuentran algunos componentes electrónicos, como son resistencias, condensadores, algunos transistores, y un par de *chips.* Uno es un cuádruple amplificador operacional, el **LM 324,** muy popular entre los aficionados a la electrónica. La mayoría de los componentes se ocupan del preproceso de la señal de audio, de naturaleza analógica. El otro *chip,* tal vez el más importante en esta aplicación, es un popular conversor Analógico/Digital, que convierte las variaciones continuas del sonido en números sucesivos. Tomando muestras de la señal a intervalos de tiempo regulares, refleja su valor instantáneo en forma de un número de 8 bits (comprendido entre 0 y 255). El *chip* en cuestión es el **ADC 0804.** El sofisticado *software* hace todo lo demás.

EN MARCHA

El *software* venía en *diskette,* en la versión que **INPUT** revisó. Se incluyen varios programas de utilización. Por razones obvias, el primero que cargamos fue el titulado DEMO, que después de saludarnos en nombre de **Covox** (el fabricante), nos mostró un sencillo menú de utilización que permite grabar voz, reproducirla y obtener efectos de eco y distintas velocidades de reproducción. La tecla f1 da paso a que pronunciemos una frase. ¿Cuál? Pues la tan manida: *«Mi taylor is rich»,* que popularizó un machacón curso de inglés. Presionamos f3 y se deja oír claramente y de inmediato *«MI taylor is rich».* Veamos qué pasa con f5... *«Mi taylor is rich, Mi taylor is rich, bla, bla...»,* hasta que la frase se desvanece, cada vez más tenue. A continuación, f7 reproduce lo del famoso sastre a diversas velocidades: la primera suena como cuando ponemos un disco de 45 r.p.m. a 33 y la última parece pronunciada por los pitufos refiriéndose a la fortuna de su sastre.

UTILIZAR EL CHIP

Llegados a este punto, toca hablar del versátil *chip* **SID,** que dota al **C-64** de sus increíbles facultades musicales. La voz digitalizada controla a dicho *chip* de cara a reproducir el sonido grabado. Es por ello factible incluir nuestra voz en un programa y que pueda ser reproducida en otro **64,** sin necesidad de emplear nuevamente el módulo. Otro programa, titulado «RATE-FILTER-MOD», nos permite alterar los parámetros de reproducción. Podemos elegir la frecuencia a la que se muestrea el sonido, también las características de respuesta del filtro del **SID,** desde la anchura de banda hasta el tipo de filtro, la resonancia o la desactivación del mismo.

El ajuste del nivel de volumen recogido por la entrada del micrófono puede realizarse con mayor comodidad cargando el programa LEVEL, que visualiza la amplitud del sonido en tiempo real.

Asimismo, están incluidos dos programas de aplicación. El primero es una calculadora parlante, que realiza las cuatro reglas básicas. El otro es un reloj que nos permite ir diciendo la hora con nuestra voz; algo así como el **093** en casa.

EL SOFT ES IMPORTANTE

Dependiendo de si queremos fundir programa y voz o simplemente trabajar con la segunda, podemos utilizar los programas de base VM o VM2. VM reserva 32 Kbytes de memoria para guardar voz, mientras que VM2 solamente toma 8 K y deja los 32 K restantes para los programas en BASIC. No obstante, el manual facilita los mapas de memoria en cada una de las opciones, para que sea más fácil el trabajo del programador.

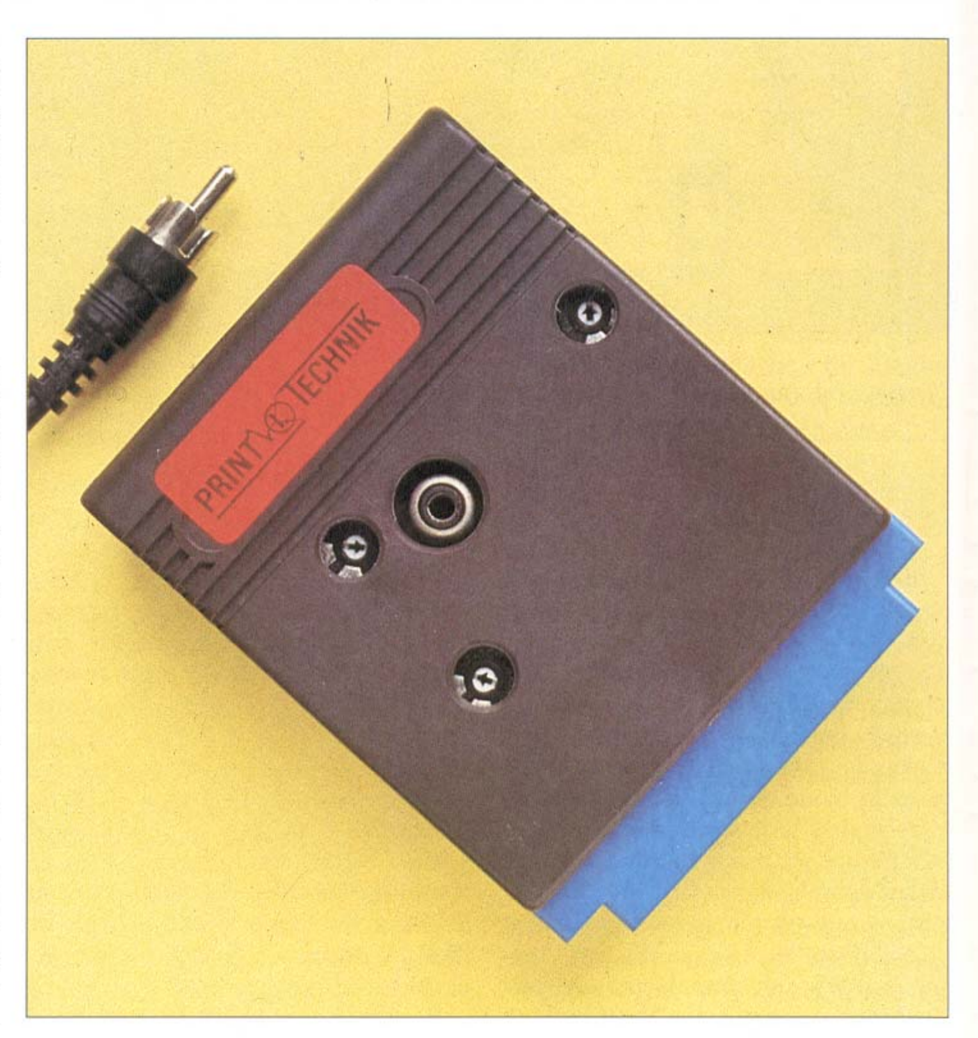
La carga del **Voice Master** en el **C-64** añade nueve comandos nuevos al repertorio del ordenador. En el cuadro adjunto se muestra su función. Una versión más reciente del **Voice Master** permite que el ordenador reconozca palabras o frases mediante un *software* que recurre a un algoritmo de reconocimiento de patrones.

VERLO PARA DIGITALIZARLO

Reza un antiguo proverbio chino que una imagen vale más que mil palabras. En la actualidad esto puede seguir siendo cierto; pero ahora, con el Video Digitalizador **de la firma austriaca** Print Technik, **una imagen puede ser cuantificada en 32.000 puntos (o *pixels*), estando el** Commodore 64 **de por medio.**

El producto que describiremos a continuación viene compuesto por un cartucho enchufable en el *port* del usuario, situado en la parte posterior del **64.** El complemento obligado es el *software* incluido en dos *diskettes,* el segundo de los cuales contiene una atractiva y cuidada demostración de las asombrosas posibilidades del digitalizador.

El ojo observador que facilita la conexión con el mundo exterior puede ser cualquier fuente que proporcione una señal de video más o menos estática, que permanezca detenida durante unos cuantos segundos.



El cartucho del ***Digitalizador*** *se conecta directamente al* port *del usuario.*

PRIMERO LA CAMARA

Lo más práctico es una cámara de video, de las habitualmente empleadas para circuito cerrado de televisión, pero tal vez sea más cómodo un reproductor de *videocassettes* normal dispuesto en «imagen fija» durante unos segundos. Es seguro que la cinta sufre algún desgaste en estas condiciones, pero la imagen es absolutamente estable, porque la cabeza de lectura lee una y otra vez el mismo campo (una imagen de las 25 por segundo que puede reproducir).

Más o menos en el centro de la superficie del cartucho asoma un conector coaxial destinado a recibir el cable, también coaxial, que trae la señal. Lo único que debe tenerse en cuenta es que la fuente de la señal de vídeo se ajusta al clásico estándar de 1 voltio pico a pico (la amplitud máxima de la tensión de la señal) y que la impedancia sea de 75 ohmnios. Esto es algo que no debe preocuparnos, pues es bastante corriente que los fabricantes de cámaras y vídeos adopten esta norma tan extendida. En otro caso, un simple adaptador de impedancias soluciona el problema.

UN CARTUCHO PARA TODO

En la misma superficie del cartucho se observan fácilmente tres orificios redondos, por los que asoman las ranuras destinadas a mover el cursor de otros tantos potenciómetros ajustables. Las ranuras admiten la punta de un atornillador del tamaño adecuado y permite girar el cursor a lo largo de un recorrido de 270 grados, hasta encontrar el punto óptimo en cada caso. Este tipo de potenciómetros ajustables son similares a los empleados para el control de volumen de los receptores de radio o televisión, pero carecen de eje, que es sustituido por el destornillador al que aludíamos previamente. Se da la



PROGRAM-MENU ▲

VIDEO-DIGITIZER IN ACTION:

curiosa circunstancia de que, a pesar de la procedencia austriaca del cartucho, los potenciómetros son del fabricante español **Piher.**

A lo que íbamos: Cada uno de los tres potenciómetros permite controlar el contraste, el brillo y la anchura de la imagen, respectivamente. No hay normas fijas, y el usuario debe calibrarlos por el método de intento y fallo, hasta encontrar que la imagen digitalizada corresponde a su gusto. El potenciómetro de ajuste de anchura de la imagen (el más próximo al ordenador) puede hacernos pensar que el dispositivo no funciona si no fue convenientemente ajustado —al menos eso nos pasó a nosotros—.

La puesta en marcha es tremendamente sencilla. En primer lugar se enchufa el módulo siguiendo la habitual recomendación para este caso: el ordenador no debe recibir alimentación eléctrica durante la conexión o desconexión del cartucho. Después se inserta el *diskette* en la unidad y se teclea LOAD "*",8. De todas maneras el fabricante dispone del *software* en *cassette* o en otro cartucho como opciones. El resultado de la carga del *software* es una pantalla-menú como la mostrada en la foto que ilustra a este artículo. Una mano con el dedo índice estirado (un *sprite*) puede ser desplazado a nuestra voluntad por la pantalla bajo el control de las telcas de movimiento del cursor. Una vez que el dedo señala nuestra elección, la tecla RETURN lo ejecuta.

En un primer momento, la opción que se nos antojó más atractiva es Digitize; movemos la manita por la pantalla y pulsamos suavemente la tecla RETURN. La pantalla es desactivada durante unos pocos segundos, que el fabricante asegura en 5, pero en nuestro caso fueron algunos más, aunque esto no sea algo demasiado importante.

La primera imagen obtenida no es probable que sea atractiva. Es cuestión de paciencia al regular los mandos, tanto los del cartucho como los que lleva la cámara en sí.

EXPLORACIONES DE LA IMAGEN

Durante el tiempo en que la pantalla ha permanecido desactivada, el **C-64** ha explorado la imagen, a su vez explorada por la cámara. Este proceso se realiza como si de un barrido se tratara, el ordenador interpreta la imagen que aparecerá en la pantalla siguiendo un desplazamiento en líneas verticales, una tras otra, desde el lado izquierdo de la pantalla hasta llegar al derecho. Sin embargo, memoriza una imagen de dimensión mayor que la realmente visualizada en los 160 por 200 puntos de que es capaz la pantalla. Notemos que se ha elegido el llamado modo gráfico por mapa de bits multicolor, que permite emplear hasta cuatro colores diferentes en cada célula de 8 por 8 *pixels* (para más detalle, ver algún libro o artículo que describa los modos gráficos del **64**). La dispo-

nibilidad de este número de colores reduce a la mitad la resolución horizontal del ordenador, que es de 320 por 200 en el modo que emplea solamente dos colores.

Volviendo a la exploración de la imagen. El **Vídeo Digitalizador** divide la imagen inicial en una rejilla de 256 por 256 puntos, 65.536 en total. Como vimos, solamente se pueden visualizar 160 por 200, que es una ventana con respecto a la imagen total. Sin embargo, esta ventana puede ser desplazada a nuestra voluntad, con las teclas de cursor y CLR/HOME, hasta centrar la zona de la imagen que más nos agrade.

La imagen es digitalizada en función de una escala de cuatro grises o, mejor dicho: blanco, negro y dos niveles de gris. Ahora bien, aprovechando que trabajamos en modo multicolor, podemos asociar un color distinto a cada uno de los niveles, obteniendo así un coloreado artificial de la imagen. El proceso es bien sencillo, cada una de las cuatro teclas de función del **64** está asociada a un nivel y cada vez que presionamos una de ellas, el gris que le corresponde se torna en un nuevo color. El resultado solamente depende de nuestro criterio artístico. La tecla que lleva el signo de la libra nos retorna al mundo del blanco y negro.

OPCIONES IMPORTANTES

El resultado obtenido puede ser almacenado en *diskette* para su utilización posterior. Unicamente hay que desplazar el dedo índice hasta la casilla SAVE. Pero existe un par de



Digitalización coloreada.

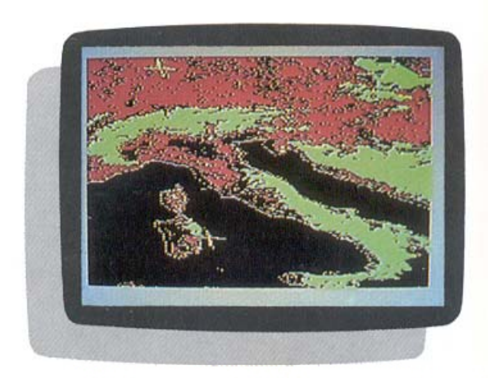
interesantes opciones, consistentes en guardar la imagen siguiendo el formato adoptado por los paquetes de *software* para gráficos **Koala** y **Paintmagic.** Por ejemplo, se podría optar por el **Koala;** después cargar ese *software* en el ordenador, conectar la **Koala Pad** al ordenador y efectuar retoques una vez cargada la imagen digitalizada. ¿Suena atractivo, verdad?

Mientras decidimos qué imagen digitalizada es la que nos gusta, podemos repetir el proceso simplemente con presionar la barra de espacios, después de haber efectuado los retoques en los mandos del cartucho y/o la cámara.

Desde el menú podemos visualizar el directorio del *diskette* que deseemos, sin necesidad de desprendernos del programa **Vídeo Digitalizador,** el movimiento del dedo lo permite, igual que la salida al BASIC directamente, sin necesidad de poner a cero el ordenador, conservándose la información relativa a la imagen, que podría ser empleada en el programa en que estemos trabajando.

La utilización de un lápiz óptico también está observada en el menú. Por si todo lo anterior fuera poco, las imágenes pueden ser transferidas a un trozo de papel, utilizando directamente una impresora. La gama donde elegir es nutrida: desde las clásicas de **Commodore,** hasta la **Canon 1210, GP 700, Okimate 20** —obteniéndose incluso reproducciones en color—, sin olvidar a las **Epson** o **STAR.** Resulta interesante recalcar que cuando se dispone de una impresora con *interface* paralelo tipo **Centronics** se puede hacer una conexión directa partiendo del *port* del usuario, sin necesidad de otra cosa que los dos conectores adecuados y un metro de cable plano. El manual proporciona el diagrama con las conexiones adecuadas.

Una de las opciones del menú está estrechamente relacionada con la impresión en color. Es la que aparece con la leyenda "16 Colors", que discrimina hasta 16 niveles de gris en la figura original, a pesar de solamente poder visualizar hasta 4 en la pantalla (debido a las limitaciones del **64** en este sentido). Sin embargo, se puede controlar una impresora en color para que haga uso de esta posibilidad. De todas formas, lo bien cierto es que para obtener una buena digitalización la imagen captada por la cámara debe serlo con la mayor calidad posible. La conexión previa de la cámara directamente al monitor, para comprobar que se cumplen estos requisitos, demostró ser un buen recurso. La digitalización se hace sobre un área de menor tamaño que la total captada por la cámara.



Digitalización coloreada.

APLICACIONES

El manual es breve, cuatro páginas y una cubierta. Es sucinto y conciso, pero tal vez una aplicación tan interesante debería ir acompañada por descripciones más detalladas.

Al fabricante no le falta imaginación cuando describe las posibles aplicaciones del **Vídeo Digitalizador,** desde vigilancia de recintos por exploración continua y comparando cada sucesiva imagen digitalizada, hasta aplicaciones médicas tales como conteo de glóbulos o bacterias (con la ayuda de un microscopio), sin olvidar el reconocimiento de imágenes o la creación de motivos para discotecas. Incluso describe la curiosa utilidad que ha encontrado un peluquero, quien digitaliza la efigie de sus clientes y les dibuja el peinado con el lápiz óptico hasta que el parroquiano está de acuerdo con su nuevo *look*.

ESTE ES EL SIMBOLO DE COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.

Este PC de Commodore ha hecho dudar a más de uno.

La oferta del mercado de los ordenadores PC dejaba hasta hoy muy sencilla la elección. Sin embargo, Commodore, líder reconocido en varios sectores de la informática, ha ofrecido una respuesta alternativa que atiende plenamente las exigencias empresariales y de profesionales liberales: su nuevo ordenador PC.

El nuevo Commodore PC dispone de una versatilidad acorde con una tecnología depurada en constante evolución y compatible con el software standard que más le suena.

commodore PC

Estos avances, y un precio realmente interesante, han planteado serias dudas entre los profesionales más cualificados a la hora de elegir un buen PC.

Sin duda Commodore, con el mayor número de ordenadores vendidos en el mundo se afianza en el campo empresarial con mucha fuerza.

Si está interesado en conocer más de cerca el nuevo PC de Commodore, pregunte en cualquier concesionario Commodore, le sacará de dudas.

PRINCIPALES CARACTERISTICAS

– 256 K de RAM de 9 bits – Zócalos para 512 K más – 2 unidades de diskete de 360 K – Disco 10 Mb opcional – Interfases serie y pararelo, incluidos – 5 slots compatibles – Alta resolución incluida – El mejor precio en esta categoría.

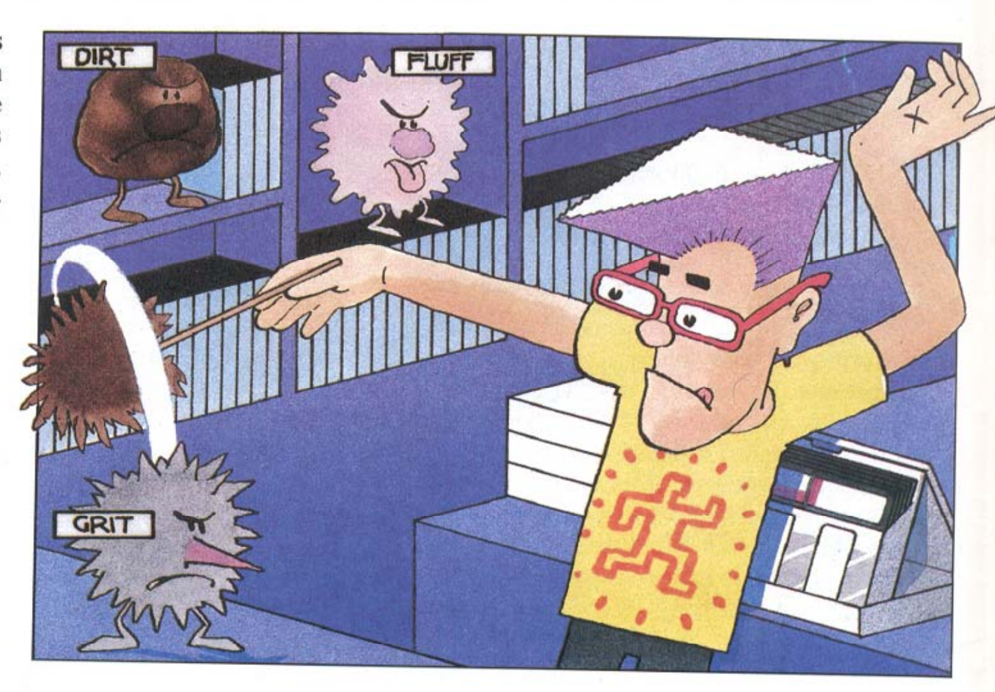


Microelectrónica y Control *c/ Valencia, 49-53 08015 Barcelona - c/ Princesa, 47 3.º G 28008 Madrid*
Unico representante de Commodore en España.

CUIDADO DE CINTAS Y DISKETTES

- PROTEGER LA INFORMACION
- HACER COPIAS DE SEGURIDAD
- INDEXAR
- ENVIO POR CORREO CINTAS Y DISCOS

Incluso si tú eres una de esas personas que guardan su información importante en la parte posterior de un sobre ya utilizado, cuando guardes ésta en cinta o *diskette,* desde luego, no te merecerá la pena ser desorganizado.



Los sistemas de almacenamiento basados en cintas o discos magnéticos hacen posible que el usuario de ordenadores personales pueda guardar grandes cantidades de información, o miles de programas, de una manera extremadamente compacta.

Pero la eficiencia que proporcionan los sistemas de almacenamiento magnético es también un potencial punto débil. Debido precisamente a que una diminuta cinta o *diskette* pueden contener tanta información, cualquier daño que se les produzca se convierte en un resultado desastroso. Aquí, el soporte magnético es un material particularmente vulnerable.

SEGURIDAD

Obviamente, lo más importante es asegurarse de que una vez depositada la información en la cinta o *diskette* ésta permanecerá allí, pudiendo ser recuperada siempre que lo deseemos. Pero existen dos clases de daños que pueden afectar a este tipo de soporte: el físico y el magnético.

El daño físico puede ser cualquier cosa, desde doblar un *diskette* o triturar una cinta, hasta acumular suciedad en su superficie. Esto se previene fácilmente, protegiéndolos siempre que no se estén utilizando. Se deben guardar en su sobre e introducirlos en una caja adecuada para *diskettes.* Las cintas también deberán meterse dentro de sus cajas y, preferiblemente, emplear uno de esos módulos de plástico que hay para apilar un buen número de cajas. Finalmente, cabe recomendar su almacenamiento lejos del calor, la humedad y el polvo.

Tampoco se debe tocar la superficie magnética de un *diskette* o el trozo de cinta que asoma por un *cassette.* También es un buen hábito rebobinar las cintas hasta su comienzo, una vez que hemos acabado de utilizarlas; esto significa que dejamos al descubierto la parte menos vulnerable de la misma. Asimismo logramos que la cinta esté disponible para su inmediato uso en cualquier momento. No es conveniente dejar olvidados los *diskettes* y las cintas en sus correspondientes unidades durante largos períodos de tiempo después de utilizar el ordenador.

Existen determinados campos magnéticos que tienen fuerza suficiente para alterar el contenido de una cinta o *diskette.* Estos proceden de diversos electrodomésticos habituales en el hogar. Por ejemplo, los altavoces contienen potentes imanes, y algunos motores eléctricos también. Por lo general, cintas y *diskettes* se deben mantener bien alejados de cualquier equipo eléctrico, incluyéndose la televisión.

MAS SEGURIDAD

El accidente puede tener lugar, por lo que merece la pena hacer copias de seguridad o tomar algún tipo de precauciones extraordinarias, especialmente para las grabaciones más importantes.

Por lo general, es una buena idea grabar dos versiones de todo, incluso en la misma cinta o *diskette.* Existen menos probabilidades de que se deterioren dos que una sola. Pero si el fichero es realmente importante haz otra copia en distinto lugar. Se almacenará en otro sitio y preferiblemente no se utilizará. Otro consejo es guardar en cinta copias de los programas existentes en *diskette* —la cinta es menos delicada.

Proteje tus cintas contra la sobreescritura quitando las lengüetas

posteriores del *cassette*. Los *diskettes* pueden ser protegidos igualmente con sólo pegarles una etiqueta adhesiva en las muescas laterales de la funda.

SIGUE LA PISTA

Las cintas y *diskettes* pueden haberse utilizado hace mucho tiempo, por lo que resultará difícil recordar, con exactitud, lo que hemos depositado en ellos. Una regla general consiste en etiquetar las cosas tanto como podamos. Si tu economía te lo permite, no está de más almacenar cada fichero por separado, en cintas cortas, o reservando *diskettes* para ficheros relacionados entre sí.

Proporciona a cada fichero un nombre claro y único (dentro de los límites que permita tu sistema). En particular, si tienes varios desarrollos de un mismo programa, dale a cada uno un nuevo nombre (o número). Escribe los nombres de los ficheros



en la etiqueta del *cassette,* así como en la tarjeta que va en la caja, pues ambas deben ir asociadas en caso de separación. No escribas en las etiquetas de un *diskette* con un lápiz duro porque te puedes cargar el *diskette*. Es preferible emplear un rotulador de punta de fieltro. Un libro que actúe como índice de ficheros será de gran ayuda. En él puedes indicar todos los nombres, dónde está el fichero y cualquier otra anotación. Merece la pena incluir una sentencia tipo REM en el fichero, en la que se anota la fecha y descripción.

ENVIO DE FICHEROS

Las cintas y *diskettes* son una manera muy conveniente de enviar información o programas por correo. Las cintas son razonablemente fuertes y viajan bien. Sorprendentemente, lo delicado y más propenso a rotura es la cajita, por lo que si la evitas, además ahorrarás peso. Aunque difíciles de encontrar, existen carteritas para envío por correo, pero otro tipo de bolsas almohadilladas son perfectamente adecuadas.

El mayor riesgo que corren los *diskettes* es el de ser doblados. Por ello un bocadillo hecho con dos láminas fuertes evita este problema. Es muy importante disponer un mensaje en parte visible del paquete, en el que se lea algo así como: MATERIAL MAGNETICO, MANEJESE CON CUIDADO. Para mayor seguridad, el correo certificado suele ser una buena solución.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** *que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podríamos haber recurrido a otros métodos de muestreo, pero éste no deja lugar a dudas.*

El resultado de las votaciones será publicado en cada número de ***INPUT.***

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: *No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.*

Enviad vuestros votos a:

LOS MEJORES DE INPUT
Alberto Alcocer, 46-4.° B
28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido

....

Segundo título elegido

....

Tercer título elegido

....

Programa que te gustaría conseguir

....

¿Qué ordenador tienes? ....

Nombre ....

Apellidos ....

Dirección ....

....

Teléfono de contacto ....

INPUT COMMODORE

LOGO, LENGUAJE INTUITIVO

Cada vez son más los niños que se introducen en la informática a través del LOGO, un lenguaje diseñado pensando en el aprendizaje. Seymour Papert, **a quien se atribuye un claro conocimiento de la mentalidad infantil, ha introducido una nueva revolución en las aulas con su desarrollo de un lenguaje de alto nivel.**

En 1960 los ordenadores eran muy caros. La potencia de cálculo de tu micro habría costado millones de pesetas, ya que incluso los ordenadores más grandes no podían almacenar más de unos 144 Kbytes. Por razones de economía, los lenguajes de ordenador estaban diseñados para utilizar la menor cantidad de memoria posible, y su concepción pretendía hacerlos fáciles para el ordenador aunque así resultaran más difíciles para el programador.

Con la aparición del microordenador en los años setenta, los lenguajes de programación ganaron en popularidad, ya que los nuevos micros, al igual que los grandes ordenadores de los años sesenta, tenían memorias pequeñas. Cobró cuerpo la idea de que «sencillo para el ordenador» no significaba necesariamente «sencillo para el programador», y se fueron aceptando las dificultades de aprendizaje de lenguajes como el BASIC, como una característica de la programación.

LA ELECCION DE UN LENGUAJE

Al conectar la mayoría de los micros domésticos, éstos trabajan en BASIC (Beginners' All-Purpose Symbolic Instruction Code=Código de Instrucciones Simbólicas de Uso General para Principiantes), siendo éste el lenguaje que domina la mayoría de los propietarios de micros. Sin embargo no hay razón para que esto sea así necesariamente. El BASIC es sólo un programa en código máquina que está presente automáticamente en el ordenador, y de hecho aún existen máquinas para uso doméstico en las que el BASIC ha de cargarse desde una cinta o disco antes de empezar a programar.

Esto significa que es perfectamente posible cambiar el lenguaje que tu ordenador es capaz de entender; todo lo que tienes que hacer es cargar un programa un código máquina que le permita reconocer las instrucciones y realizar las acciones adecuadas. Tal vez ya has descubierto que es relativamente fácil ampliar el BASIC de tu máquina con comandos adicionales.

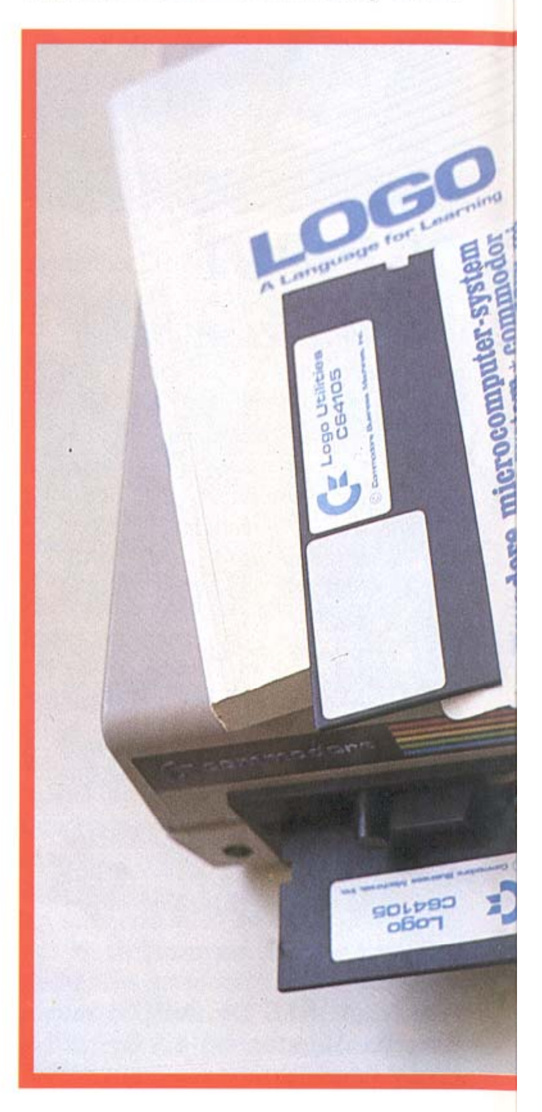
Pero también es posible no adaptarse meramente al programa existente utilizado por el intérprete BASIC, sino sustituirlo completamente.

Desde los primeros tiempos de la aparición de los ordenadores, se han desarrollado alrededor de unos cien lenguajes diferentes, para diversos fines, además de los múltiples lenguajes "de fabricación casera" diseñados para ordenadores particulares. Algunos de estos lenguajes son tan especializados en sus aplicaciones que nunca te encontrarás con ellos, excepto en los más altos niveles de investigación. Pero hay otros que por lo menos son tan prácticos para el usuario doméstico como el BASIC, y en algunos casos incluso más.

Que puedas disponer o no de un lenguaje particular depende de que puedas obtener o no el programa que permite a tu micro operar con él. Los diferentes lenguajes se suministran normalmente como cualquier otro programa, sobre cinta o disco, y en algunos casos sobre un *chip* de ROM. Su disponibilidad depende del ordenador que tengas. Por ejemplo, muchas máquinas de aplicación comercial que disponen del sistema operativo CP/M pueden elegir entre más de una docena de lenguajes, y en algunos casos entre varias versiones de cada uno. Y existe también una elección relativamente variada para la mayoría de los ordenadores domésticos.

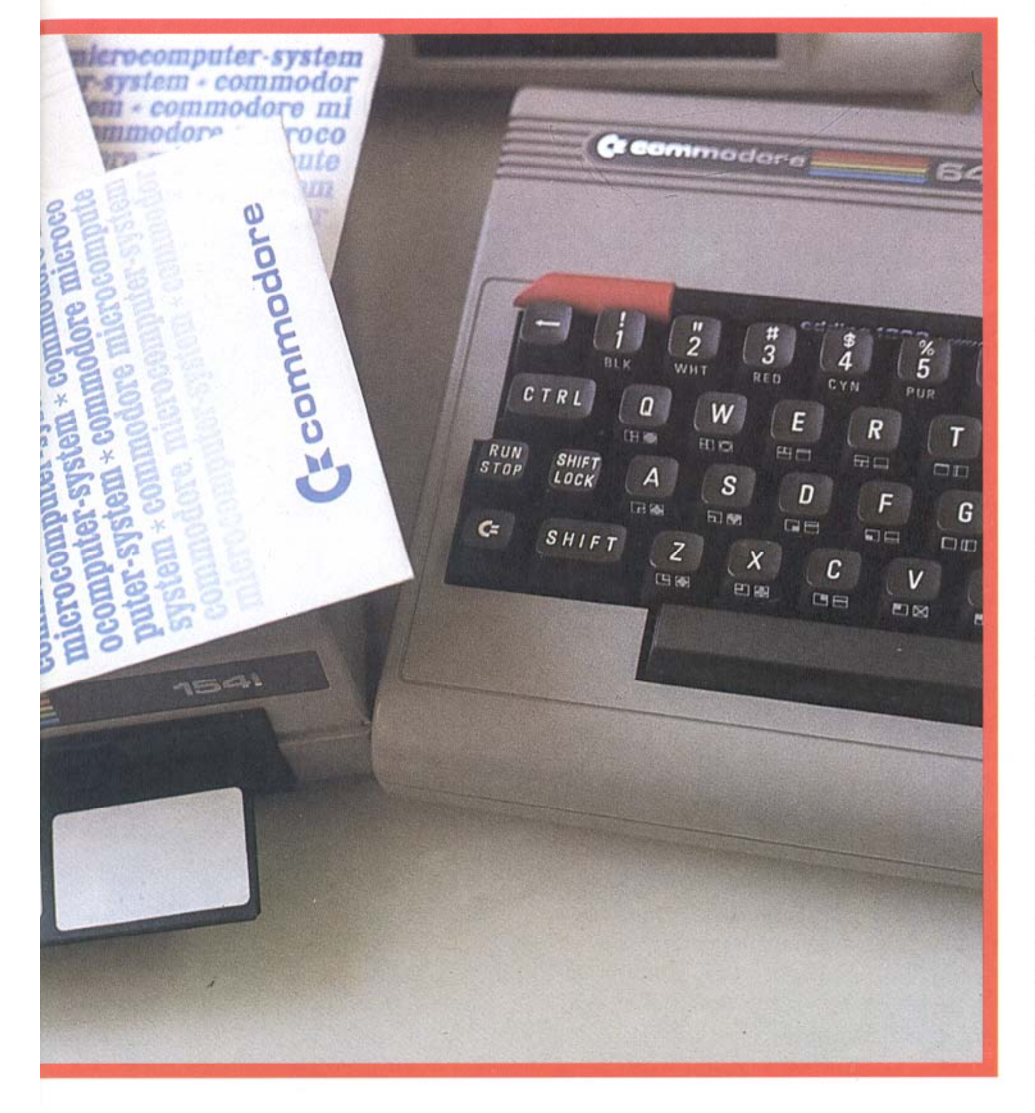
NIVELES DE COMUNICACION

Un lenguaje es una forma de comunicar entre el ordenador y tú. Es



algo que entendéis los dos, un compromiso entre un lenguaje natural (por ejemplo el inglés y el código máquina en binario, que es con lo que realmente trabaja la máquina). Se dice que un lenguaje es de bajo nivel cuando es cercano al lenguaje propio del ordenador. Un ejemplo de ello es el lenguaje ensamblador. Los lenguajes de alto nivel, como el LOGO, pueden estar muy cerca del lenguaje natural. La próxima generación de ordenadores (llamada quinta generación) probablemente utilizará uno de dichos lenguajes para aceptar instrucciones directamente del inglés.

El BASIC se sitúa en algún punto intermedio entre los dos extremos, y según la opinión de muchos programadores no se puede decir que sea un buen compromiso, ya que ni es fácil de entender, ni es rápido para ser usado por el ordenador.



LOS COMIENZOS DEL LOGO

En 1967, un grupo de investigación del **Instituto Tecnológico de Massachusetts** (MIT), estableció unos planteamientos nuevos de acercamiento a los ordenadores. Se propusieron la creación de un lenguaje que resultara fácil para el programador aunque no fuera tan fácil para el ordenador. El resultado fue el LOGO.

El equipo estaba encabezado por **Seymour Papert,** un sudafricano expatriado. **Papert** había trabajado estrechamente con **Jean Piaget,** el famoso psicólogo infantil, según el cual los niños sólo pueden entender un concepto abstracto si se les presenta de una forma concreta. Pensaba que el aprendizaje de un niño debería tener lugar a través de sus propios descubrimientos, en lugar de adoptar una actitud pasiva ante las cosas. Estas ideas tuvieron una gran influencia en el desarrollo del LOGO.

También ejerció influencia el trabajo de **Mary Minsky,** investigadora sobre inteligencia artificial en el mismo instituto durante los años 60. La inteligencia artificial es la ciencia que se ocupa de simular con máquinas los aspectos de la inteligencia humana. Los ordenadores no son inteligentes, sino que sólo obedecen instrucciones que se les han de dar de una forma muy detallada. Los factores que hay que considerar cuando se resuelve un problema son muchos y muy variados, de modo que la simulación de un proceso de este tipo en un programa de ordenador es una tarea gigantesca. El mundo de la inteligencia artificial necesita lenguajes de programación que permitan simular el aprendizaje humano y su capacidad de tomar decisiones. El LISP es un potente lenguaje de programación desarrollado con tal objeto, y del que se hablará más adelante en INPUT. El LOGO es esencialmente un dialecto del LISP, y aunque puede manejar palabras y números, está orientado principalmente a la programación de gráficos.

El nombre de LISP deriva de «List Processing». Su estructura de datos básica no es una matriz de números ni una cadena de caracteres, sino una lista, y dado que una lista puede estar formada por símbolos o por otras listas, es fácil procesar datos no numéricos. Sin embargo el LISP no es fácil de aprender.

DEMOSTRACIONES CONCRETAS

Seymour Papert y sus colegas iban buscando una "puerta de acceso" por la que los niños pudieran entrar al mundo de la programación. Se identificaron así tres áreas de interés para los niños: gráficos, música y robótica. Los niños están interesados en realizar dibujos sobre el monitor

de su ordenador, en servirse de él para crear sonidos electrónicos, y en controlar máquinas desde el teclado. De las tres áreas, la robótica era la más excitante y sugestiva, lo que llevó a **Papert** a crear la tortuga robótica.

El desarrollo de la tortuga se vio estimulado porque en aquella época no había monitores baratos. La tortuga, un robot que se arrastra por el suelo llevando un lápiz, está controlada por el ordenador. El lápiz puede subirse o bajarse a medida que la tortuga se va moviendo, con lo que ésta puede realizar dibujos. Permite que los niños relacionen la geometría con sus propios movimientos al andar o pintar. Con frecuencia se veía a los niños que utilizaban la tortuga moviéndose y probando subrutinas antes de ordenar que lo hiciera la tortuga.

La tortuga se llamó así en honor de **Grey Walter,** un neurólogo y cibernético británico que construyó "tortugas cibernéticas" en los años 50. Se trataba de vehículos accionados eléctricamente que medían el nivel de carga de sus baterías, y cuando estaban bajas se dirigían hacia un dispositivo de carga al que se enchufaban ellas solas. La "tortuga" de **Grey Walter** fue uno de los primeros robots verdaderos.

La tortuga robótica original del MIT casi se extinguió con la aparición de los ordenadores personales y la facilidad de representar cosas en una pantalla de una forma sencilla y barata. Fue sucedida por una versión bidimensional: la tortuga de la pantalla. Se trata de un cursor, representado a veces como un *chevron* y otras veces como una pequeña tortuga. Obedece a las mismas instrucciones que la tortuga robótica, no se rompe y es mucho más barata.

Sin embargo la tortuga robótica ha hecho su reaparición en las escuelas, ya que permite que participen más niños en las actividades de programación, proporcionando un modelo para la «geometría de cuerpos», y una excitante introducción concreta a un mundo abstracto.

Estos robots se están haciendo muy populares en las escuelas elementales y están cubriendo su objetivo original de introducir a los niños en la programación de ordenadores de una forma divertida y comprensible.

HABLANDO EL LENGUAJE

¿Qué tienen que ver los juguetes mecánicos con un lenguaje de ordenador? **Papert** considera el ordenador como un vehículo para la creatividad y la expresión de ideas. Piensa que la mejor manera de aprender cosas acerca de los ordenadores es crecer en una cultura del ordenador, de la misma forma que la mejor manera de aprender italiano es pasarse una temporada en Italia. En la conferencia anual correspondiente a 1983 de la principal asociación profesional de ordenadores de América, pidió que se buscase una fórmula para darle un ordenador a cada niño americano. Piensa que es el niño quien debe programar al ordenador y no el ordenador quien programe al niño. Los medios que propone para lograr esto pasan por el LOGO, que según su punto de vista da a los niños control sobre uno de sus más potentes recursos, proporciona una base de partida para resolver problemas fuera del cálculo, y permite que se presenten de una forma sencilla e inteligible las ideas matemáticas complejas.

Papert explicó su filosofía en su célebre libro "Frenesí mental, Niños, Ordenadores e Ideas Potentes". Desde la publicación de su libro en 1980, han aparecido versiones del LOGO para la mayoría de los ordenadores domésticos. Está disponible para diversos microordenadores y la mayor parte de las versiones se parecen estrechamente al LOGO original del MIT. También hay en el mercado varios programas con nombres tipo LOGO, como el «Logo Dart» y el «Logo Graphics».

Se trata de simulaciones de la tortuga gráfica, que sólo constituye una pequeña parte del LOGO. El lenguaje tiene utilidades completas para el proceso de listas y palabras, funciones matemáticas y de sonido y muchas otras prestaciones que no aparecen en los programas "Pseudo-Logo".

El LOGO es el primer lenguaje "próximo al usuario". Como es sencillo de aprender y abunda en los colegios, se piensa con frecuencia que es "para chicos". Nada más lejos de la verdad. El profesor **Harold Abelson,** uno de los diseñadores del LOGO en el MIT, declara: "Al trabajar con el LOGO hemos descubierto algunas cosas importantes."

"Un lenguaje de ordenador puede ser sencillo y potente al mismo tiempo. De hecho no se trata de aspectos conflictivos sino complementarios, ya que es la falta de potencia expresiva en los lenguajes primitivos tales como el BASIC, lo que hace tan difícil para los principiantes el escribir programas simples que hagan algo interesante. Lo que es más importante, hemos visto que es posible dar a la gente control sobre potentes recursos informáticos, que pueden usar como herramientas aprendiendo, jugando y explorando."

El LOGO es también un lenguaje que crece. La universidad de Edimburgo está desarrollando una versión llamada Control-LOGO, que permita un control más sofisticado de los robots. **Papert** pretende que las futuras versiones incluyan "mundos" en los que los niños puedan jugar con las ideas de la Física, de la misma forma que juegan con la Geometría con la Tortuga Gráfica.

PROGRAMANDO EN LOGO

Al cargar el LOGO desde un *cassette* o un *chip* de ROM, te enviará un mensaje de este tipo:

```
BIENVENIDO AL LOGO

?
```

La interrogación (?) es un anunciador o invitación a que introduzcas algo. El LOGO está esperando a que se le dé alguna orden. Si tecleas:

```
HOLA LOGO, TE HE ESTADO
  BUSCANDO
```

El LOGO responderá con:

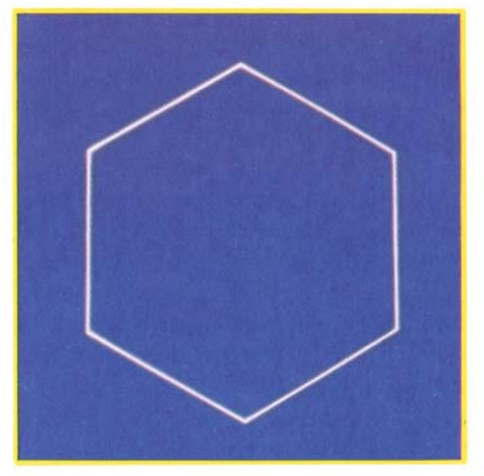
```
NO SE COMO HACER HOLA
```

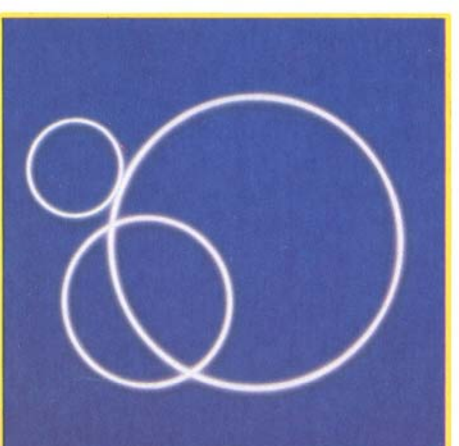
?

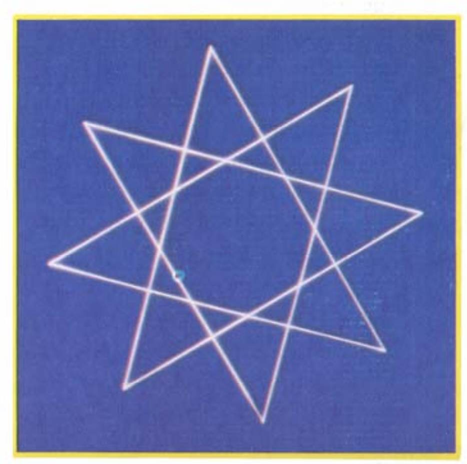
Ha mirado la primera palabra sin identificarla como instrucción, y te ha informado de ello.

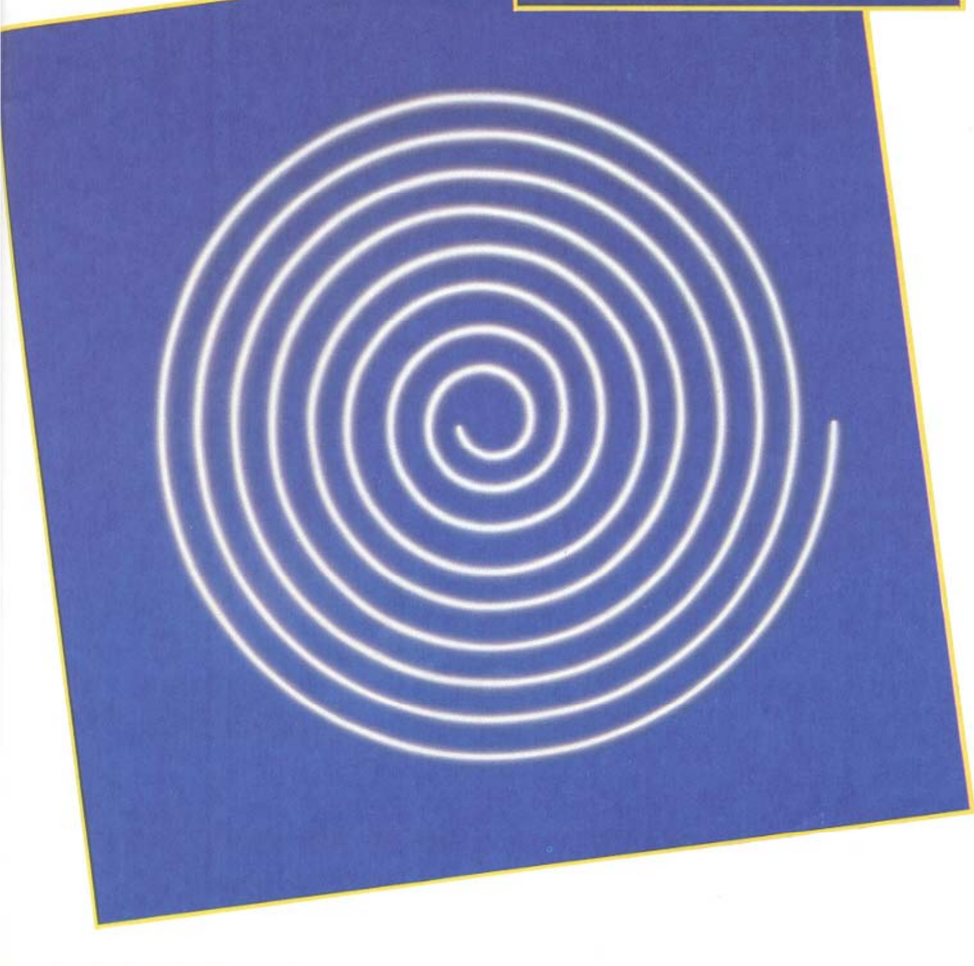
Un comando que sí reconocerá es ST, que quiere decir mostrar la tortuga (Show Turtle). Llamará a la tortuga desde las profundidades del ordenador hasta la pantalla. La forma de la tortuga aparecerá en una dirección particular. Este es su encabezamiento, aparecerá provista de un lápiz y listo para pintar.

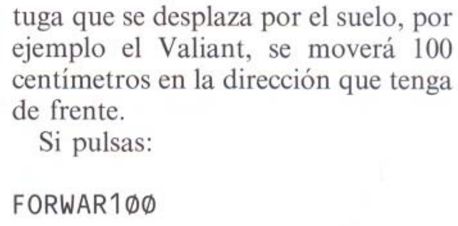
La instrucción para mover la tortuga hacia adelante es FORWARD, que se puede abreviar a FD. FORWARD es una instrucción del LOGO que requiere una entrada. Hay que decirle a la tortuga cuánto tiene que moverse. Si se pulsa FORWARD 100, la tortuga se moverá 100 unidades hacia adelante. Dejará tras de sí una línea. Si está utilizando una tor-











tuga que se desplaza por el suelo, por ejemplo el Valiant, se moverá 100 centímetros en la dirección que tenga de frente.

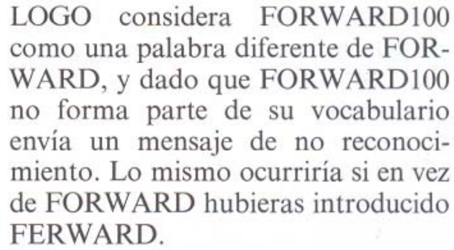
Si pulsas:

```
FORWAR100
```

El LOGO responderá

```
NO SE COMO HACER FORWAR100
```

Esto se debe a que, al no haber intercalado un espacio entre el comando y la cantidad numérica, el



LOGO considera FORWARD100 como una palabra diferente de FORWARD, y dado que FORWARD100 no forma parte de su vocabulario envía un mensaje de no reconocimiento. Lo mismo ocurriría si en vez de FORWARD hubieras introducido FERWARD.

Hay un sencillo editor de línea que permite corregir los errores antes de pulsar RETURN. Las teclas del cursor permiten trasladarse a lo largo de la línea hasta llegar a la falta. La tecla de borrado (delete) elimina el carácter situado a la izquierda del cursor. Para insertar nuevos caracteres no tienes más que teclearlos. El texto que hay más a la derecha se moverá automáticamente para dejar sitio al texto insertado. Algunas versiones de LOGO permiten recuperar y editar una línea después de haber pulsado RETURN.

La instrucción BACK trabaja de la misma forma que FORWARD. Su abreviatura es BK. Hace que la tor-

tuga se mueva hacia atrás, y, como antes, requiere una entrada numérica. Puedes dar cualquier número como entrada para FORWARD o BACK. Así cambiará la posición de la tortuga, pero no su orientación.

Para girar la tortuga, utiliza LEFT y RIGHT. Se pueden abreviar por LT y RT. Al igual que FORWARD y BACK, requieren una entrada numérica. Por ejemplo, RIGHT 39 hará girar la tortuga 39 grados a la derecha. Análogamente LEFT 123, hará que la tortuga gire 123 grados hacia la izquierda.

ALGUNOS JUEGOS SENCILLOS

Si se utiliza el LOGO con niños, es mejor no decirles que RIGHT 96 hace girar la tortuga 96 grados hacia su derecha. Hay que dejar que experimenten con diferentes valores numéricos de entrada y descubran por sí mismos el efecto de los diferentes valores. Se pueden ensayar diversos juegos encaminados a que descubran los valores de diferentes ángulos.

Se puede marcar una posición en la pantalla y ver cuántas instrucciones necesita un niño para detener la tortuga debajo de la marca. O se puede dibujar una carretera en la pantalla y hacer que el niño haga avanzar la tortuga, perdiendo un punto por cada instrucción y un punto cada vez que la tortuga se salga de la carretera. Más adelante daremos en este artículo un procedimiento para dibujar una carretera.

El programa se puede almacenar en disco o cinta, y cargarse cada vez que los niños quieran jugar con él. Si quieres jugar inmediatamente, dibuja la ruta directamente sobre la pantalla del televisor, utilizando un lápiz de cera, o un rotulador de los usados para pintar sobre pizarra blanca. Se puede borrar con facilidad al terminar. De esta forma es más fácil dibujar rutas más complicadas.

Hay una variedad de juegos que se pueden practicar con una tortuga de suelo, basados en recorrer laberintos, tirar cosas, recoger objetos en diversos puntos, etc. Se puede jugar a la tortuga empujona, una variante del juego de los bares, que utiliza una tortuga en vez de una moneda.

Es posible jugar a la tortuga empujona en la pantalla, escribiendo un procedimiento para dibujar la tortuga, y colocándola al principio.

MAS COMANDOS

Para jugar con las instrucciones de traslación y giro se requieren algunas otras instrucciones de LOGO o **primitivas.** Si la pantalla no está en modo "enrollado", es decir, si la tortuga desaparece por la parte superior pero no reaparece por la inferior, necesitas hacer que vuelva a aparecer.

HOME hará que la tortuga regrese a su posición y orientación originales en el centro de la pantalla.

También necesitas poder borrar la pantalla para hacer nuevos dibujos.

La instrucción CLEARSCREEN, que se abrevia CS, borrará todos los dibujos y situará la tortuga en su posición HOME.

A veces desearás que la tortuga se mueva sin dibujar. La instrucción PENUP, abreviada PU, levanta el lápiz de la tortuga. Si utilizas una tortuga de suelo, se levanta el lápiz que lleva en la panza. La tortuga de la pantalla simplemente deja de dibujar.

La instrucción PENDOWN, abreviada PD, hace que baje el lápiz de la tortuga de suelo y que la tortuga de pantalla dibuje de nuevo.

Las instrucciones CLEARSCREEN, HOME, SHOWTURTLE, PENUP y PENDOWN son primitivas de LOGO que no requieren entradas.

He aquí un ejemplo ilustrativo sobre estas primitivas. Al final de cada línea has de pulsar la tecla RETURN.

```
SHOWTURTLE
LEFT 45
FORWARD 71
RIGHT 135
PENUP
FORWARD 50
PENDOWN
LEFT 45
BACK 71
PENUP
HOME
PENDOWN
```

Si quieres intentar alguna otra cosa, puedes borrar el dibujo de la pantalla utilizando la instrucción CLEARSCREEN.

CONSTRUCCION DE UN PROCEDIMIENTO

Hasta aquí todas las actividades descritas han sido del modo 'inmediato". Has estado hablando directamente a la tortuga, que ha ejecutado inmediatamente tus órdenes, igual que un pelotón de soldados obedeciendo a un sargento instructor en una plaza de armas. Hay otro modo de funcionamiento en LOGO, el modo de procedimiento.

En el modo de procedimiento se nombra una serie de comandos, los cuales se escriben después del nombre. De esta forma el nombre asignado al procedimiento pasa a formar parte del vocabulario del LOGO. La tortuga responderá a este nombre ejecutando los comandos que figuran en su definición. Para definir un procedimiento, utiliza TO seguido del nombre elegido. Se puede utilizar cualquier nombre excepto el de una primitiva de LOGO existente. Las cosas resultan más fáciles si asignas a los procedimientos nombres que definan su función.

Se entenderá mejor esto si ponemos un ejemplo. He aquí un proeedimiento que enseña a la tortuga a dibujar un zig-zag:

```
TO ZIGZAG
FORWARD 20
LEFT 150
FORWARD 20
RIGHT 150
FORWARD 20
LEFT 150
FORWARD 20
RIGHT 150
FORWARD 20
LEFT 150
FORWARD 20
RIGHT 150
END
```

Después de teclear TO ZIGZAG seguido de RETURN la invitación o *prompt* cambia de ? a >. Esto te indica que estás en modo procedimiento. Al terminar el procedimiento, pulsa END y el anunciador volverá a ser ?, lo que significa que nuevamente estás en modo inmediato.

ZIGZAG forma ahora parte del vocabulario del LOGO. Si lo tecleas, la tortuga dibujará un zig-zag.

El LOGO tiene una instrucción de repetición que se puede utilizar para teclear varias veces la misma cosa. Se podría escribir de nuevo ZIGZAG como:

```
TO ZIGZAG
REPEAT 3 [FORWARD 20 LEFT 150
  FORWARD 20 RIGHT 150]
END
```

La rutina que ha de repetirse aparece encerrada en paréntesis cuadrados, precedida por REPEAT y el número de veces que ha de ser repetida. Si piensas en un círculo como en una serie de traslaciones y giros cortos, REPEAT facilita el dibujo de curvas. Por ejemplo REPEAT 180 [FORWARD 1 RIGHT 1] dibujará un semicírculo.

PONIENDOLO TODO JUNTO

Ya tienes todos los ingredientes para dibujar una pista de carreras. La manera de resolver un problema con el LOGO es dividirlo en lo que **Seymour Papert** llama "Bytes a medida de la mente". Así, para empezar con el borde interior, dibuja primero una curva:

```
REPEAT 180 [FORWARD 1
RIGHT 1]
```

Y ahora una línea recta:

```
FORWARD 100
```

Puedes combinar dos curvas y dos bordes rectos en un procedimiento para dibujar el interior de la pista:

```
TO INTERIOR
REPEAT 2 [FORWARD 100 REPEAT
  180 [FORWARD 1 RIGHT 1]]
END
```

Es perfectamente legítimo tener en REPEAT dentro de otro REPEAT siempre que te acuerdes de cerrar todos los paréntesis al final.

Para el exterior de la pista, necesitas una curva más amplia, creada aumentando el tamaño de los pasos de la tortuga.

```
REPEAT 180 [FORWARD 2
  RIGHT 1]
```

es demasiado grande, pero algo cercano a:

```
REPEAT 90 [FORWARD 3 RIGHT 2]
```

estará bastante bien.

Vayamos ahora al exterior de la pista.

```
TO EXTERIOR
REPEAT 2 [FORWARD 100 REPEAT
  90 [FORWARD 3 RIGHT 2]]
END
```

Pero si tecleas:

```
EXTERIOR
INTERIOR
```

no tendrás el resultado ideal.

Puedes escribir un procedimiento para empezar a dibujar en un sitio más adecuado, mediante un procedimiento llamado BEGIN:

```
TO COMENZAR
PENUP
LEFT 40
FORWARD 110
RIGHT 130
PENDOWN
END
```

Otro procedimiento moverá la tortuga a una posición adecuada para dibujar la parte interior de la pista, proporcionando además una línea de partida.

```
TO MOVER
RIGHT 90
FORWARD 30
LEFT 90
END
```

Para ver la pista debes llamar a todos los procedimientos a la vez, tecleando:

```
COMENZAR
EXTERIOR
MOVER
INTERIOR
```

Ahora puedes escribir un procedimiento que coloque a la tortuga en la línea de salida:

```
TO PRINCIPIO
LEFT 90
FORWARD 15
RIGHT 90
END
```

Una vez que un procedimiento está ya en la memoria de un ordenador, el LOGO te permite usarlo de la misma forma que cualquier primitiva de LOGO. Esto significa que puedes usar los procedimientos para ayudar a definir otros nuevos procedimientos. Así, puedes combinar los bloques constructivos que has llamado BEGIN, OUTSIDE, MOVE, INSIDE y START, en un nuevo procedimiento que denominaremos GAME:

```
TO JUGAR
COMENZAR
EXTERIOR
MOVER
INTERIOR
PRINCIPIO
END
```

Cada vez que tecleas GAME, aparecerá la pista con la tortuga en la línea de salida.

Todos los programas de LOGO se construyen de esta forma. La fragmentación de un programa en pequeños bloques constructivos hace más fácil su construcción y depuración.

En un próximo artículo veremos cómo realizar programas más complicados utilizando las técnicas aquí descritas.

COMO TRABAJA EL SISTEMA OPERATIVO

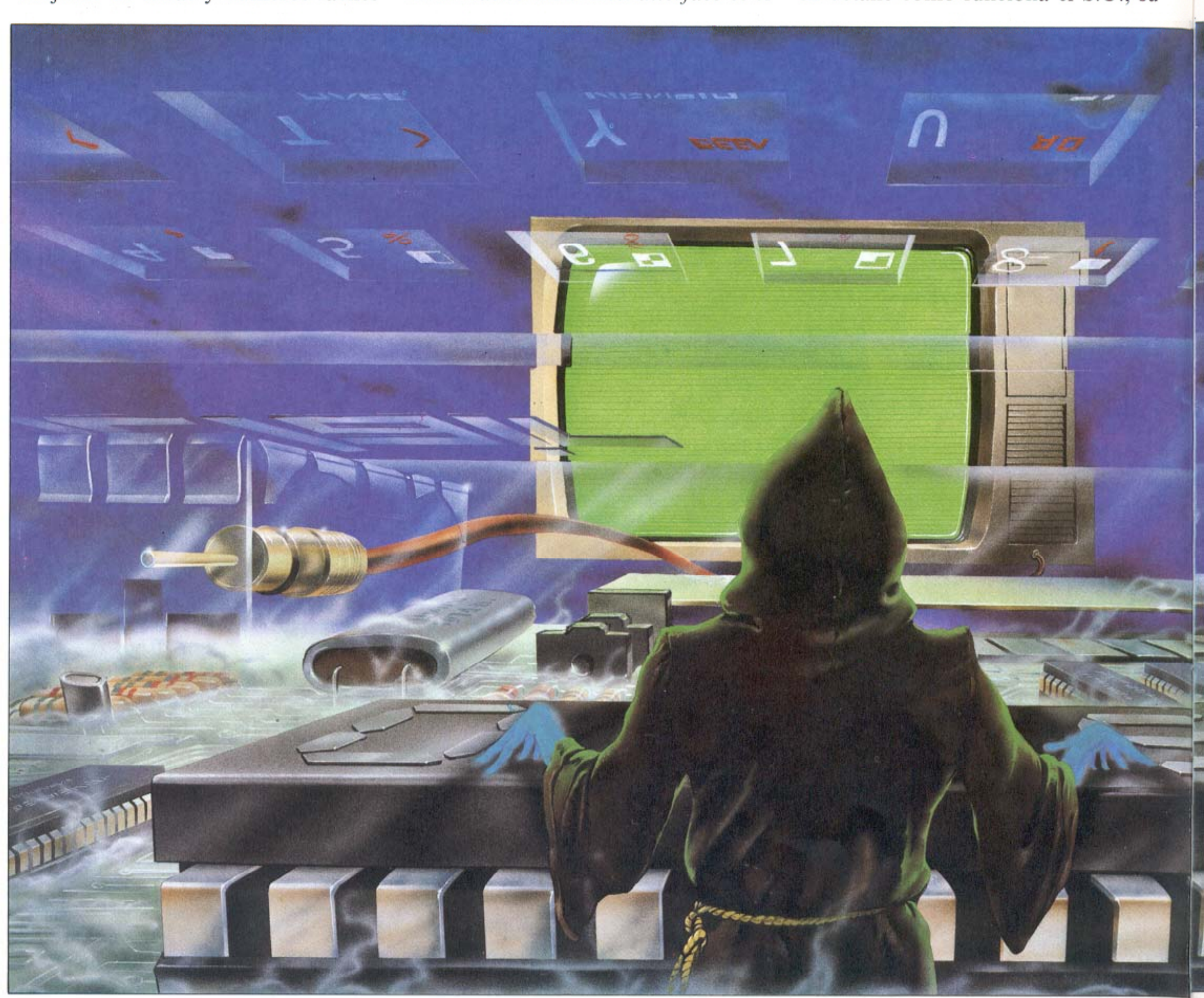
- ¿QUE ES EL SISTEMA OPERATIVO (S.O.)?
- TRABAJAR POR ENCARGO
- ENTENDER EL BASIC
- RUTINAS DE INTERES

Todos los ordenadores trabajan y entienden un solo lenguaje: el código máquina. Consiste éste en una interpretación instantánea de las variaciones de tensión eléctrica en base a las cuales opera el ordenador.

El usuario puede representarlo de una forma sencilla, con un simple conjunto de letras y números fáciles de recordar, o de una forma complicada, con una lista de ceros y unos casi incomprensible. Cualquiera que sea la forma en que esté representado, pocas personas pueden trabajar cómodamente con el código máquina, resultando necesario un medio de enlace o *interface* con el ordenador, que permita la comunicación a un nivel más sencillo. Este *interface* es el Sistema Operativo (S.O.), un paquete de *software* que gobierna las funciones del ordenador y hace posible que te comuniques con la máquina.

Independientemente del lenguaje que utilices, es seguro que éste se servirá de un sistema operativo para controlar al ordenador.

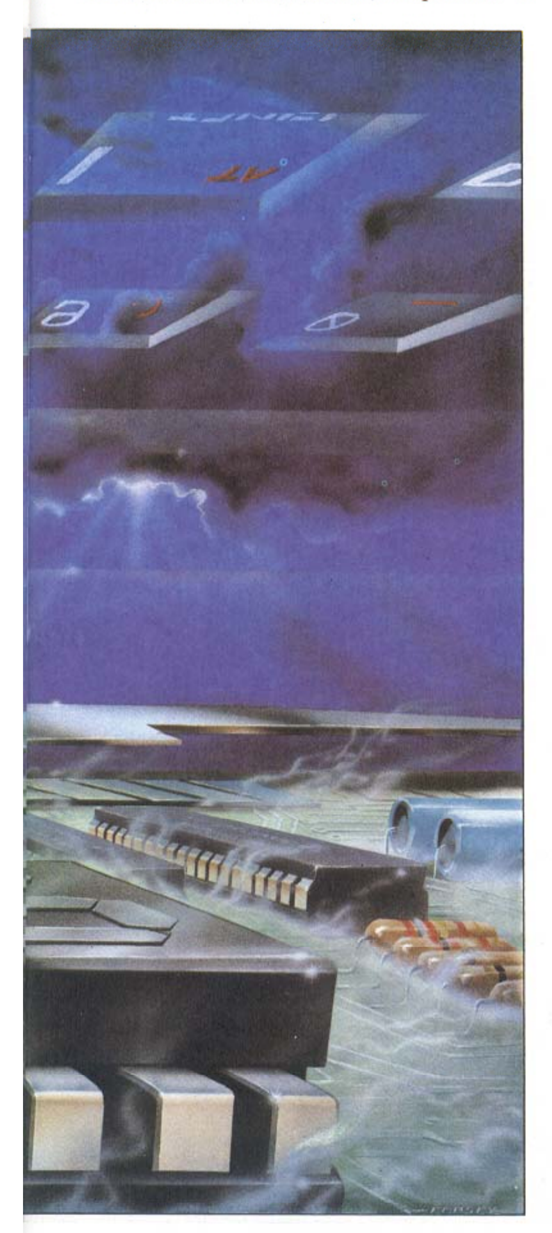
Aunque no es necesario entender en detalle cómo funciona el S.O., su



conocimiento te permitirá aumentar tu habilidad como programador y obtener mayor provecho del microordenador. Hasta dónde penetrar en el S.O. depende del diseño de tu ordenador. Dentro de estos límites, este artículo pretende familiarizarte con el S.O. de tu modelo y mostrar cómo puede utilizarse parte de la amplia colección de rutinas de actuación rápida que incorpora.

¿QUE ES EL S.O.?

El S.O. no es más que un sofisticado programa en código máquina que permite al procesador, incluido en un determinado ordenador, responder a tus instrucciones. Controla las interacciones de la máquina con el mundo exterior, la pantalla, el teclado, el generador de sonidos y otros *ports* de Entrada y Salida. El S.O. garantiza también que las posiciones de memoria están adecuadamente asignadas y que el procesador y el usuario utilizan la memoria de una manera eficiente. Cuando se conecta el ordenador, el S.O. inicializa ciertas rutinas de la ROM y carga los registros internos y los punteros con determinados valores iniciales, todo ello antes de enviar al usuario la señal de que está preparado para seguir adelante. En el caso de los microordenadores como del que tratamos aquí, el sistema operativo se ocupa normalmente de manejar los comandos del BASIC, pero también puede manejar otros lenguajes.



La mayor parte de la actividad de un S.O. está relacionada con el ajuste y comprobación de punteros. Por ejemplo, es importante que el procesador sepa dónde empiezan y terminan los programas en BASIC. Cuando se está editando el programa, va cambiando su tamaño y los punteros han de ser actualizados.

El mismo principio se aplica al ajuste y mantenimiento de espacio para almacenamiento de variables y, especialmente, de cadenas alfanuméricas. Cuando se DIMensiona una cadena hay que reservar una zona de memoria para ella y posteriormente ha de borrarse cuando la cadena ya no se necesite. Es esencial que el S.O. gestione la memoria de una forma eficiente, de lo contrario los programas o cadenas largos ocuparán toda la memoria. Esta gestión de memoria se llama mantener la casa en orden, y una buena gestión de memoria es una de las características que distinguen a un ordenador eficiente.

El S.O. incorpora una serie de rutinas de *software* que han sido convenientemente dispuestas para poder ser reclamdas por el programador o por el propio S.O. El programador de BASIC no necesita la mayoría de estas rutinas, ya que las funciones que realizan están contenidas en instrucciones normalmente introducidas cuando se utiliza el BASIC. Un buen ejemplo de esto es la palabra clave INPUT del BASIC. Esta palabra realmente utiliza varias subrutinas del S.O., incluyendo entrada, ajuste del canal, exploración del teclado y transferencia del *buffer*, pero el usuario del BASIC puede reclamarlas a todas utilizando una sola instrucción.

Otro ejemplo de una subrutina del S.O. reclamada desde el BASIC es la sentencia CLR del **Commodore.** Con ella se obliga al S.O. a que realice toda la gestión interna que sea necesaria. Su efecto es dejar disponible para el programador toda la parte de RAM que se ha utilizado, pero que ya no se necesita, borrando todas las variables.

TRABAJANDO POR ENCARGO

Cada vez que se da una instrucción al ordenador, el S.O. activa a la sección adecuada del mismo. Por ejemplo, si se pulsa una A en el teclado, es el S.O. quien da instrucciones al microprocesador para que saque una A por la pantalla. Para hacer esto, detecta que se ha pulsado una tecla. En algunas máquinas, entre ellas el **Commodore,** el S.O. explora el teclado regularmente para detectar las pulsaciones de las teclas. En otras máquinas, cada pulsación de tecla envía una señal de interrupción al S.O., que explora entonces el teclado para localizar la tecla que ha sido pulsada.

Para reaccionar ante la pulsación de una tecla, el S.O. va a la subrutina que se utiliza para sacar por pantalla un carácter. Esta es una de las muchas subrutinas en código máquina que se ejecutan muy rápidamente. Si se pulsa la tecla RETURN el S.O. se dirige a una subrutina de "nueva línea" y la ejecuta, disponiendo al cursor en el principio de una línea nueva.

ENTENDER EL BASIC

Cuando se ejecuta un programa (RUN) en un lenguaje, tal como el

BASIC, las instrucciones almacenadas en la memoria son interpretadas, es decir, traducidas a código máquina, que es lo que entiende el ordenador. Al mismo tiempo, el código ajusta los registros para que se pueda acceder a las subrutinas del S.O. De esta forma las mismas subrutinas que permiten que el ordenador responda a las entradas introducidas por el usuario a través del teclado, por ejemplo, son utilizadas por el S.O. para ejecutar los programas en BASIC.

¿Entonces por qué —podríamos preguntar— resulta el BASIC tan lento comparado con el código máquina, habida cuenta de que utiliza estas subrutinas rápidas? La respuesta es que el lenguaje ha de ser interpretado a partir de palabras clave, tales como PRINT y otros símbolos antes de que se pueda acceder a las subrutinas del S.O. El largo tiempo necesario para traducir el BASIC es la razón por la que muchos juegos de marcianitos y algunos programas que han de tener una respuesta rápida, se escriben en código máquina.

El tiempo que se tarda en escribir en lenguaje máquina se estima a menudo en unas diez veces el tiempo que se tardaría en escribir un programa en BASIC que hiciera lo mismo, pero el resultado puede ser cincuenta veces más rápido. Sin embargo, a menos que sea absolutamente necesario utilizar el código máquina, la mayor parte de la gente utiliza el BASIC para escribir programas.

Sin embargo sería de gran utilidad para los programadores en BASIC el poder acceder a las subrutinas del S.O. directamente, en vez de hacerlo a través del intérprete, y aprovecharse así de sus muchas subrutinas eficientes. Por desgracia, la mayoría de los micros domésticos no poseen un S.O. al que se pueda acceder directamente desde el BASIC.

Por ejemplo en los **Commodore** el S.O. incluye al intérprete BASIC, una colección de subrutinas **Kernal** y un programa **Kernal Control.**

Pese a la diversidad de accesos que presentan las distintas máquinas, hay algunos hechos que los usuarios pueden aprovechar para resolver problemas y errores.

Para tener acceso a una subrutina en ROM o del S.O., en **Commodore** se utiliza SYS seguida de la dirección en decimal.

SYS 58648 inicializa la pantalla en su resolución y color normales. Esto es más sencillo que una serie de POKEs, método usual en BASIC.

SYS 58692 (58179 para el **Vic 20**) borra la pantalla. El método usual es un PRINT seguido por el símbolo gráfico de corazón inverso, pero este signo no está disponible en algunas impresoras.

Para desplazar una línea hacia arriba lo que aparece en pantalla, introduce SYS 59626 (59765 para el **Vic 20**). Esto puede ser mucho más sencillo que los símbolos gráficos para mover el cursor.

Un arranque en frío se logra con SYS 64738 (64802 para el **Vic 20**).

INPUT JUEGOS

COLECCIONABLE DE PROGRAMACION



Aquí aprenderás cómo diseñar tus propios juegos para ordenador; desde los más simples a los más complejos de estrategia e inteligencia. Una gran abundancia de ejemplos son la manera más agradable de introducirte en los secretos de la programación.

MOVIMIENTO Y ANIMACION

- LOS PRINCIPIOS DE LA ANIMACION
- MOVIMIENTO DE GRAFICOS
- COMO UTILIZAR LOS GRAFICOS INCORPORADOS

¿Quieres darle vida a tus juegos de programación? Empieza entonces con estos sencillos caracteres gráficos, que puedes generar a partir de la memoria gráfica de tu ordenador.

Jugar con los juegos que se venden para tu ordenador es divertido sólo hasta cierto punto. Llega un momento en que la mayor parte de la gente siente la necesidad de dar rienda suelta a su imaginación y crear programas de juegos propios.

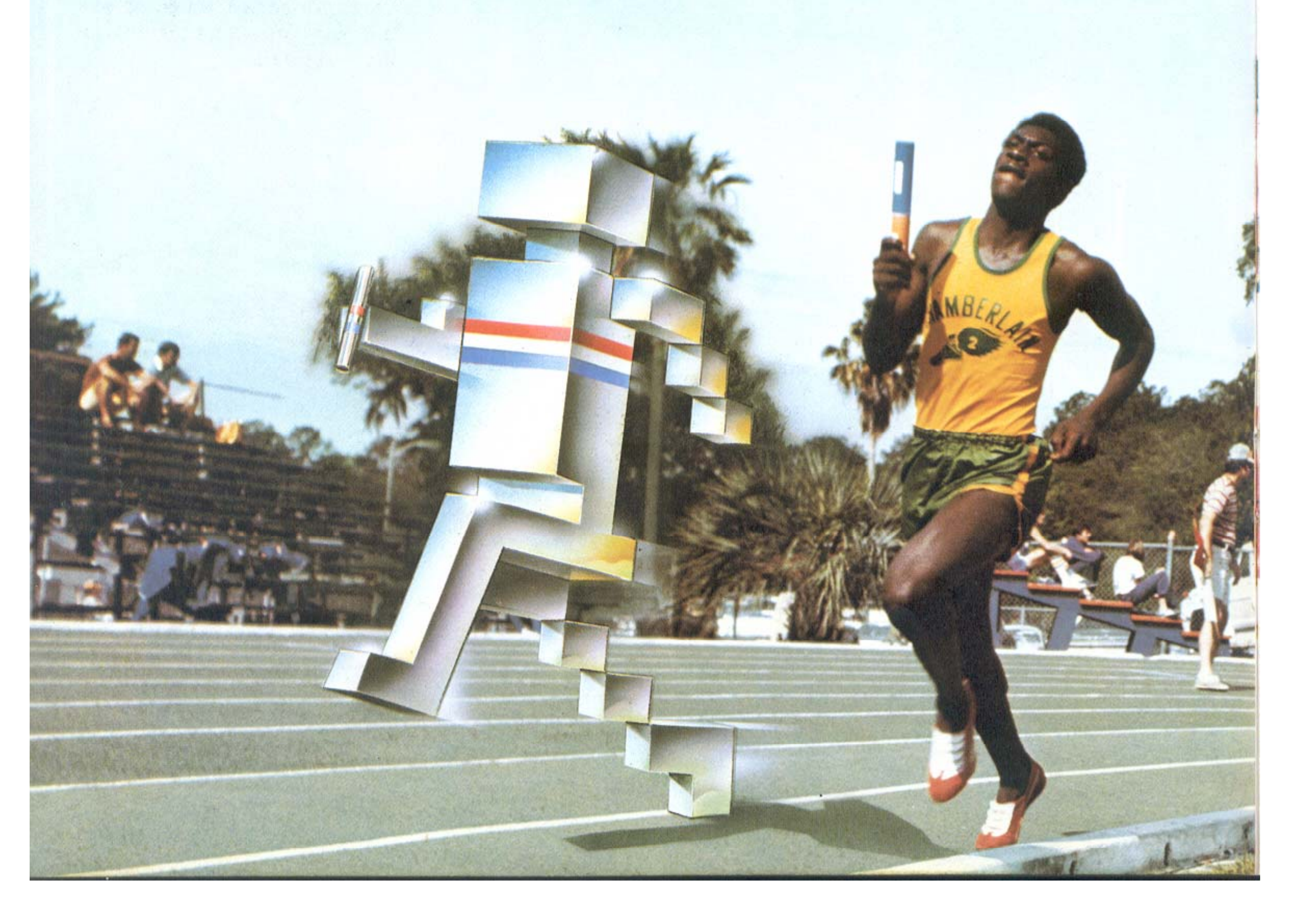
La programación de juegos no es fácil; tienes que empezar con cosas muy sencillas e ir aumentando poco a poco en complejidad. Pero eso te ayudará a pensar con lógica y aumentará tu habilidad como programador. Y también te divertiras más.

Lo primero que tienes que aprender en la programación de juegos, aparte de los trucos del BASIC, es la técnica de la animación.

Para crear la ilusión de movimiento, el programador de ordenadores utiliza en gran parte la misma técnica que el caricaturista que anima una película de dibujos. Lo que hace es crear dos (o más) imágenes y alternarlas rápidamente (idealmente, unas 24 veces por segundo).

Pero existe una diferencia importante. En la animación de dibujos, el dibujante cuenta con el proyector de películas, que le permite olvidarse de una imagen cuando ya no la necesita. En la animación por medio de un ordenador no ocurre esto. Como no hagas algo para evitarlo, cualquier segmento de imagen que "proyectes" sobre un área dada de la pantalla permanecerá allí indefinidamente.

Una forma de olvidarse de la imagen que ya no se necesita es, simple-



mente, imprimir algo sobre ella. El ordenador no puede situar dos imágenes al mismo tiempo en la misma posición de la pantalla.

Así si, por ejemplo, la línea 10 de un programa cualquiera dice al ordenador que imprima una A en una determinada posición, cualquier otra línea de programa que imprima (por ejemplo) una B en la misma posición se desembarazará de la A.

Los programas que siguen contienen varios ejemplos de esta clase de sustitución.

¿Pero qué ocurre si no tienes nada que imprimir encima del carácter no deseado? Acuérdate en ese caso de incluir en alguna línea posterior una instrucción que imprima un espacio en blanco en la correspondiente posición de la pantalla.

Si te olvidas de este detalle, tu pantalla pronto se verá abarrotada con trozos no deseados de brazos, piernas y cuerpos.

La manera de obtener los caracteres gráficos en la pantalla difiere mucho de un ordenador a otro. Hay diferencias entre los caracteres gráficos incorporados en ROM y lo mismo ocurre con la forma en que se imprimen (con PRINT) en pantalla. Por último también hay diferencias en el modo de hacer que se muevan.

Esta versión de un ciempiés para **Commodore** utiliza exactamente los mismos signos de máquina de escribir que otras máquinas, pero el método de generarlos sobre la pantalla es diferente. Intenta introducir el siguiente programa:

```
)))     )))     )))     )))
OOO<    OOO<    OOO<    OOO<
)))     )))     )))     )))
```

```
1Ø PRINT"▼"
2Ø PRINT")))"
3Ø PRINT"OOO<"
4Ø PRINT")))"
5Ø PRINT"■"
6Ø PRINT"((("
7Ø PRINT"OOO<"
8Ø PRINT"((("
9Ø GOTO 1Ø
```

Cuando lo ejecutes (RUN) observarás una imagen rápidamente cambiante. Se debe a los conjuntos separados de símbolos de las sentencias PRINT que se superponen unos a otros. Al mismo tiempo, la sentencia GOTO de la última línea crea un bucle continuo: le dice al ordenador que vuelva a empezar.

Sin las líneas 10 y 50 el programa no podría funcionar adecuadamente. Ambas hacen uso de los caracteres especiales disponibles en el modo "comillas" de las máquinas **CBM,** que te permitirán incorporar movimientos del cursor y otros controles a la sentencia PRINT.

¿Cómo funcionan? El símbolo *home* (S. inversa) de la línea 50 hace volver el cursor a la posición *home,* arriba a la izquierda de la pantalla. Esto significa que toda actividad posterior en la pantalla empieza en esta posición, de modo que todos los caracteres, desde la línea 60 en adelante se escribirán encima de los que ya estuvieran allí.



El símbolo CLEAR/HOME (corazón inverso) de la línea 10 hace algo más. Después de hacer volver el cursor a la parte superior izquierda de la pantalla, borra todo su contenido, dejándola lista para que aparezca la siguiente imagen.

MAS DESPACIO

El movimiento del insecto es hasta ahora más bien rápido y tal vez haya que retardarlo. La manera más fácil de hacer esto es utilizar un bucle FOR...NEXT. Introduce pues la línea:

```
45 FOR T=1 TO 1ØØ:NEXT
```

Cuando pulses RETURN y ejecutes (RUN) el programa, el movimiento resultará mucho más pausado. El bucle FOR...NEXT actúa como un contador, en este caso contando hasta 100, antes de que el programa vaya a la línea 50.

Puedes cambiar la duración de la pausa sin más que sustituir "100" por cualquier otro número de tu elección: cuanto más grande el número, mayor será el retardo.

Intenta también cambiar la posición del bucle de retardo FOR...NEXT a la línea 15. Dependiendo del retardo que escojas, habrá una apreciable pausa —una pantalla limpia— cuando se ha borrado una imagen pero todavía no ha sido sustituida por la siguiente. Por esta razón es mejor utilizar HOME que CLEAR/-HOME, dentro de un programa de este tipo.

Aunque la imagen ya está ahí, todavía resulta algo espasmódica, debido a que el bucle de retardo sólo actúa sobre la primera imagen. Se puede obtener un movimiento mucho más adecuado introduciendo en el programa otro bucle FOR...NEXT que actúe sobre la segunda imagen. Inserta pues:

```
85 FOR I=1 TO 5Ø:NEXT
```

Este retardo es más corto, a fin de crear un movimiento "de pierna" ligeramente irregular, pero naturalmente, puedes cambiarlo si quieres.

CARACTERES EN MOVIMIENTO

El siguiente paso es alterar el programa de forma que el "cuerpo" del insecto parezca cruzar la pantalla. En el BASIC del **Commodore** no hay una sentencia PRINT AT, por lo que hay que usar otros métodos para posicionar los caracteres.

En aplicaciones sencillas, puedes utilizar la función TAB. TAB siempre va seguido de un número entre paréntesis, por ejemplo TAB(15), que coloca el cursor en la columna 15 de la pantalla, o por una variable entre paréntesis. Además TAB siempre forma parte de la sentencia PRINT a la que se aplica.

En este caso, utiliza una variable para hacer que la posición de TAB se mueva por la pantalla. Asimismo, utiliza otro bucle FOR...NEXT:

```
1Ø FOR P=Ø TO 35
2Ø PRINT"◘"TAB(P)")))"
```



```
30 PRINT TAB(P)"000<"
40 PRINT TAB(P)")))"
45 FOR I=1 TO 45:NEXT
60 PRINT"■"TAB(P)"((("
70 PRINT TAB(P)"000<"
80 PRINT TAB(P)"((("
85 FOR I=1 TO 50:NEXT
90 NEXT P
```

Lo que has hecho es suprimir la sentencia GO TO original de la línea 90. Ahora ha sido remplazada por un bucle FOR...NEXT, que incrementa en 1 la variable P cada vez que se repite el programa. Como P forma parte de la sentencia TAB, el insecto se mueve sobre la pantalla, a razón de un paso por cada ciclo del programa.

Al ejecutar (RUN) el programa, verás que el insecto se mueve sobre la pantalla y se para al llegar al lado derecho. Para que la acción comience de nuevo necesitas agregar:

```
10 FOR P=0 TO 35
```

GRAFICOS EN ROM

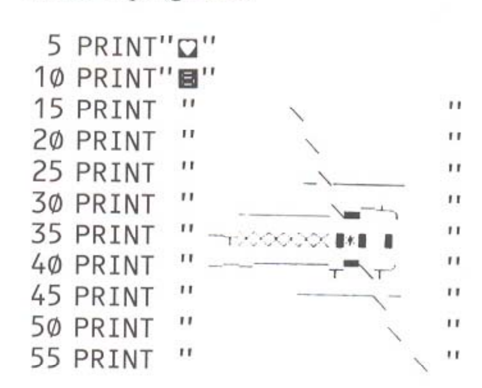
Commodore dispone de una amplia variedad de gráficos incorporados que puedes utilizar para crear formas e imágenes más elaboradas.

Para tener acceso a estas posibilidades, tienes que familiarizarte primero con el modo "gráficos y mayúsculas" que se obtiene pulsando simultáneamente las teclas [C=] y [SHIFT].

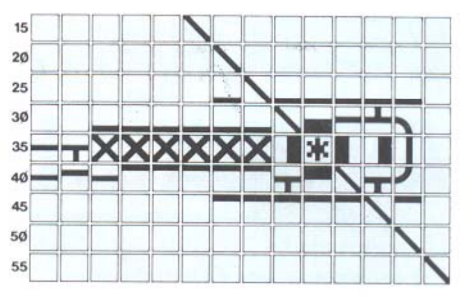
Intenta ahora teclear algunos de los caracteres gráficos. El símbolo de la izquierda se puede imprimir en pantalla pulsando simultáneamente [C=] y la tecla elegida. El símbolo de la derecha se obtiene análogamente pero pulsando al mismo tiempo [SHIFT] y la tecla elegida.

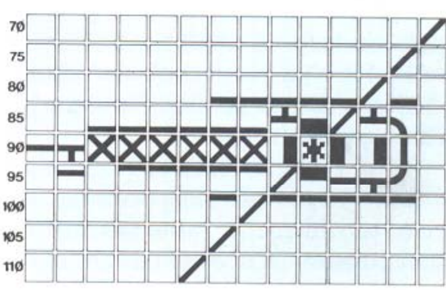
Se pueden obtener otros símbolos gráficos por medio de RVS (inverso), que se activa pulsando al mismo tiempo [CTRL] y 9, y se desactiva pulsando al mismo tiempo [CTRL] y 0.

El gráfico del helicóptero (figura 2) utiliza símbolos no inversos y da una idea de los gráficos de ROM. Aquí tienes el programa:



```
5 PRINT"■"
10 PRINT"■"
15 PRINT "                    "
20 PRINT "                    "
25 PRINT "                    "
30 PRINT "                    "
35 PRINT "                    "
40 PRINT "                    "
45 PRINT "                    "
50 PRINT "                    "
55 PRINT "                    "
```





Cómo construir un helicóptero. Cada cuadro es un gráfico de ROM.

```
60 FOR D=1 TO 50:NEXT
65 PRINT"■"
70 PRINT "                    "
75 PRINT "                    "
80 PRINT "                    "
85 PRINT "                    "
90 PRINT "                    "
95 PRINT "                    "
100 PRINT "                    "
105 PRINT "                    "
110 PRINT "                    "
115 FOR D=1 TO 50:NEXT
120 GOTO 10
```

DERECHA... IZQUIERDA... ARRIBA... ¡FUEGO!

- DETECCION DE LAS PULSACIONES DE TECLA
- LANZAMIENTO DE MISILES
- CONTROL DE UN GRAFICO MOVIL

Los juegos de extraterrestres se basan en la habilidad del jugador para controlar los sucesos de la pantalla. He aquí cómo controlar el movimiento, lanzar misiles e integrarlos en un juego.

Los juegos de marcianitos serían terriblemente aburridos si el movimiento de la base lanzadora de ráyos láser o el disparo de los mismos no se pudiera controlar de alguna manera.

El control por teclado de este tipo de movimientos es una importante faceta de todos los juegos de marcianitos, por lo que si piensas escribir alguno, es importante comprender sus principios.

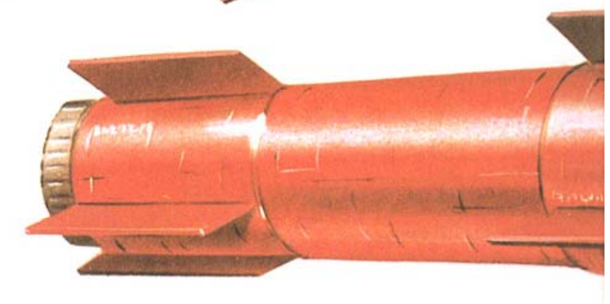
Para ello, el primer paso es hacer que el ordenador reaccione cuando pulses una tecla.

DETECCION DE LAS PULSACIONES DE TECLA

En principio todos los ordenadores domésticos se sirven del mismo método para detectar una pulsación de una tecla, si bien los detalles varían ampliamente de unos a otros.

Los ordenadores **Commodore** pueden utilizar la sentencia GET para detectar todas las pulsaciones de tecla. El siguiente programa corto muestra la utilización típica de GET para hacer esto.

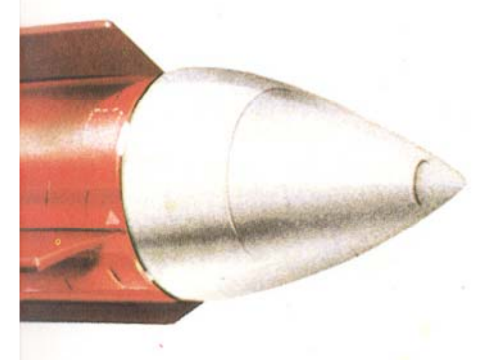
```
20 PRINT"♥"
30 GET A$:IF A$="" THEN
   GOTO 30
50 PRINT TAB(17)
   "AUUG!":
   END
```

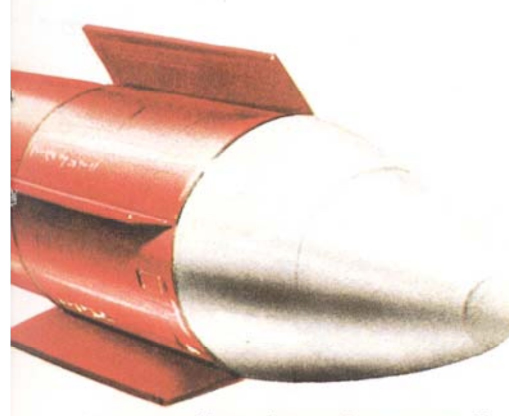


Haz correr el programa (con RUN) una y otra vez, para comprobar que todas las pulsaciones de tecla originan la aparición en pantalla del mensaje AUUG!, con excepción de las teclas RUN/STOP, SHIFT y 'Commodore C=.

La línea 20 del programa borra la pantalla. La sentencia GET en la línea 30 permite atender a las pulsaciones de tecla.







Las comillas sin nada en medio significan: «si no se ha pulsado ninguna tecla». En tal caso el programa vuelve al principio de la misma línea y espera para continuar hasta que se pulse alguna tecla.

La línea 50 presenta en pantalla el mensaje AUUG!, y el programa termina.

En un juego típico, el programa te pedirá que pulses determinada tecla para indicar tu elección a partir de un menú de selección, o para mover una rana, una base de rayos láser o lo que sea. Añadiendo una línea a tu programa, puedes especificar qué tecla ha de ser pulsada para que el programa continúe de una forma particular:

```
40 IF A$="D" THEN PRINT
   TAB(12)"MMM, ESO ESTA
   MEJOR":END
```

Cualquier pulsación de tecla conduce a la línea 40. En dicha línea se comprueba si la responsable de la pulsación fue la tecla D, en cuyo caso se ejecuta la instrucción PRINT que contiene. En cualquier otro caso, el programa salta a la línea siguiente.

LANZAMIENTO DE UN MISIL

El siguiente programa lanza un misil cuando se pulsa la tecla F. En el **Vic 20,** cambia 18 por 8 y suprime dos ▣ en la línea 30. En la línea 50, utiliza 471 en lugar de 939. En la línea 70, utiliza 7680 en lugar de 1024, 38400 en vez de 55296, y sustituye N,0 por N,1. Utiliza 7680 en lugar de 1024 en la línea 80 y 22 en vez de 40 en las líneas 80 y 90.

```
20 PRINT"♡"
30 PRINT TAB(18)"▣▣▣▣▣
   ▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣
   ▣▣▄↑▄█
40 GET K$:IF K$="" THEN
   GOTO 40
50 IF K$="F" THEN
   N=939:GOTO 70
60 GOTO 40
70 POKE 1024+N,30:POKE
   55296+N,1
80 POKE 1024+N+40,32
90 N=N-40
100 IF N>0 THEN GOTO 70
110 IF N<0 THEN GOTO 20
```

Se utilizan los gráficos de baja resolución en ROM para formar una base de misiles en el centro de la parte inferior de la pantalla (línea 30).

Las líneas 40, 50 y 60 hacen que el programa espere hasta que se ha pulsado la tecla F, disparando el misil cuando así ocurre, y dando a N un valor que es el de la posición de un misil *después* de que ha sido disparado.

La línea 70 dibuja el misil, la línea 90 lo mueve hacia arriba una línea cada vez, y la línea 80 borra la anterior posición para crear la ilusión del movimiento.

Las líneas 100 y 110 comprueban que el misil está dentro de la zona de pantalla, volviendo a empezar cuando no ocurre así.

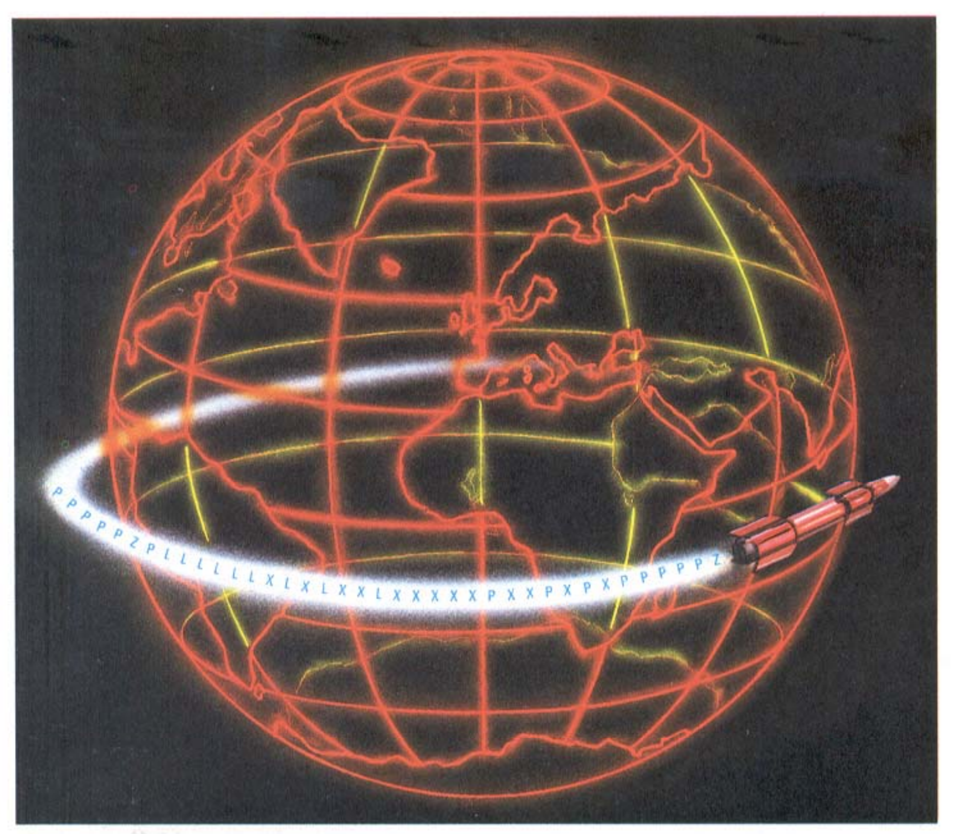
MOVIENDOSE POR LA PANTALLA

El programa de la base de misiles, tal como está, es más bien aburrido, pero si se dota a la base de movimiento, las cosas mejoran un poco. Veamos cómo puede moverse la base:

```
20 P=18
30 PRINT "♡"
40 PRINT TAB(P)"▣▣▣▣▣
   ▣▣▣▣▣▣▣▣▣▣▣▣▣▣
   ▣▣▣▄↑▄"
50 GET K$:IF K$"" THEN
   GOTO 50
60 IF K$="I" THEN P=P-
   1:GOTO 90
70 IF K$="D" THEN
   P=P+1:GOTO 90
80 GOTO 50
90 IF P>36 THEN P=36
100 IF P<1 THEN P=1
110 GOTO 30
```

En el **Vic,** utiliza 8 en lugar de 18 en la línea 20. Suprime dos ▣ en la línea 40 y pon 18 en vez de 36 en la línea 90.

La P de la línea 20 establece la posición de partida de la plataforma lanzamisiles, y la línea 40 dibuja dicha plataforma en esa posición. La rutina de pulsaciones de tecla apare-



ce en las líneas 50, 60 y 70, comprobando si se han pulsado las teclas L o R, restando 1 de P para mover la plataforma hacia la izquierda, y sumándole 1 para moverla hacia la derecha. El GOTO de la línea 80 hace que el programa vuelva a la línea 50 si se pulsó cualquier otra tecla.

En las líneas 90 y 100 se comprueba que el valor de P cae dentro de la pantalla, repitiendo el bucle GET si no ocurre así. Finalmente, la línea 110 hace que el programa vuelva a la línea base de la plataforma lanzamisiles.

CREA TU PROPIO JUEGO

Ya dispones de algunos bloques constructivos con los que puedes montarte tu propio juego. El siguiente juego te enseña una posible forma de servirte de ellos. En el **Vic** sustituye la línea 10 por

```
10 POKE 36879,29
```

Además, cambia por un 8 el 16 de la línea 40, y cambia todos los 34 de las líneas 60 y 105 por sendos 16. Por último, suprime dos ■ de la línea 50.

```
 1Ø POKE 5328Ø,5:POKE
    53281,1
 2Ø PRINT"♡"
 3Ø CLR
 4Ø LET P=16:LET A=1
 5Ø L              E
    D$="■■■■■■■■■■■■■■
    ■■■■■■■"
 6Ø LET A=INT(RND(1)*34)+3
 7Ø PRINT"■"TAB(A)♣"
 8Ø PRINT"■"TAB(P)D$
    ▁▄↑▄▁"
 9Ø GET K$:IF K$="" THEN 9Ø
 95 IF K$="Z" THEN P=P-1
1ØØ IF K$="X" THEN P=P+1
1Ø5 IF P>34 THEN P=34
11Ø IF P<1 THEN P=1
115 IF K$="F" THEN
    P1=P:D=22:GOTO 13Ø
12Ø GOTO 8Ø
13Ø PRINT"■" TAB(P1)
14Ø PRINT LEFT$(D$,D)"■■↑
    ■■□":D=D-1
15Ø PRINT"■"TAB(P1);
16Ø PRINT LEFT$(D$,D)"■■□
    ■■□":D=D-1
17Ø IF D>Ø THEN 13Ø
18Ø IF P1=A-2 THEN 2Ø
2ØØ GOTO 8Ø
```

Al pulsar RUN para ejecutar este programa, verás una estrella cerca de la parte superior de la pantalla. Las teclas Z y X trasladan la plataforma lanzamisiles hacia la izquierda y hacia la derecha, y la tecla F lanza un misil para destruir la estrella.

El programa está compuesto de tres secciones: hasta la línea 70, de la 80 a la 120, y las líneas 130 a 200.

Las líneas 130 a 200 son análogas a las del anterior programa de lanzamiento de misiles. Se han cambiado las variables y el GOTO, pero lo único nuevo es la línea 180. En esta línea se comprueba si la estrella y el misil están en el mismo sitio. Si ocurre esto, el programa vuelve a empezar.

La sección central, líneas 80 a 120, es una versión resumida del movimiento por la pantalla que vimos anteriormente.

La primera sección del programa, hasta la línea 70, realiza varias funciones. La línea 30 introduce una pequeña pausa antes de que el programa continúe. Esto es importante cuando la línea 180 completa el bucle al final del programa. Las líneas 40, 50 y 80 establecen la posición de partida de la plataforma lanzamisiles y definen su forma. En las líneas 60 y 70 se elige un lugar para la estrella y se representa allí.

MOVIMIENTOS MAS AGILES

El tener que presionar la tecla "izquierda" o "derecha" cada vez que quieres que un gráfico se mueva, es bastante laborioso. Lo que suele hacerse es construir una facilidad de auto-repetición. Para ello no hay más que utilizar un simple POKE, añadiendo la siguiente línea a tu programa

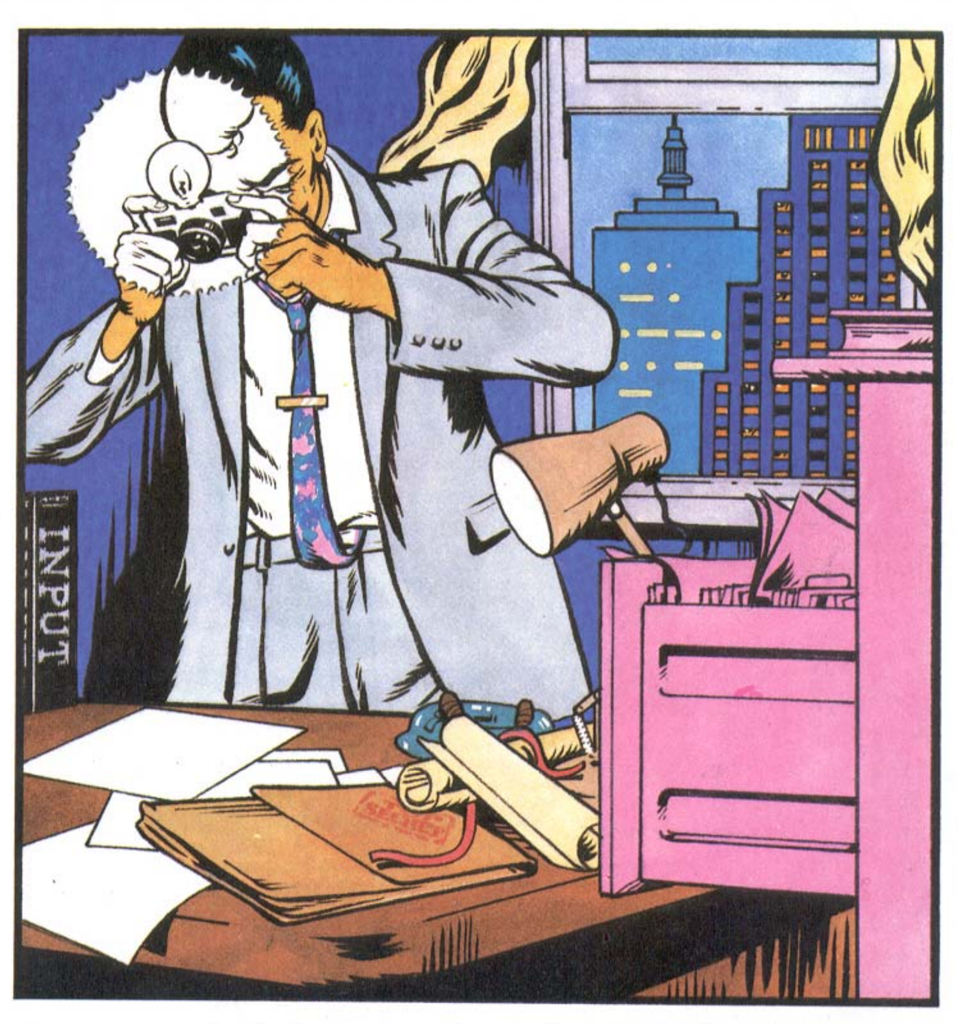
```
10 POKE 650,128
```

Puedes poner 128 o cualquier valor más alto. Para cancelar la auto-repetición, haz un POKE en la misma dirección, 650, con el valor 127.

ENVIA MENSAJES SECRETOS

- PRODUCE TUS PROPIOS CODIGOS SECRETOS
- CODIGOS DE DISTANCIA
- EL CIFRADO DE ST. CYR
- CODIGO MORSE

La codificación y el cifrado, que unen el mundo de la antigua Grecia con el mundo de la microelectrónica, son susceptibles de ser eficazmente manejados por los programas de ordenador.



La mayoría de la gente se sirve de los códigos en su vida diaria. Cuando alguien pide un tornillo del número 8 o un balón de fútbol del tamaño 5, está utilizando un código. Análogamente, las ecuaciones aparentemente incomprensibles de un científico nuclear representan la utilización de un código que permite expresar brevemente relaciones complicadas. Esto se podría hacer también utilizando palabras corrientes, pero, con frecuencia, requeriría mucho más trabajo.

En todos los anteriores ejemplos el interés reside en mejorar la comunicación de los datos, más que en buscar el secreto de la información; otra de las razones para el uso de códigos es ahorrar dinero o espacio de almacenamiento. Por ejemplo, una empresa puede redactar contratos que utilicen cláusulas y frases estándar. Si el almacenamiento se va a hacer en cinta o disco, se puede ahorrar una gran cantidad de valioso espacio codificando con números las secciones que se repiten más a menudo. A esto se le llama compresión de los datos.

Análogamente, las respuestas estándar utilizadas en juegos de aventuras, como "The Hobbit" o "Valhalla" se codifican para ahorrar espacio de almacenamiento.

Los griegos inventaron la ciencia de enviar mensajes secretos, de forma que no es sorprendente que el nombre utilizado normalmente para la codificación —criptografía— se derive de dos palabras griegas: *kryptos* (secreto) y *graphos* (escribir). Los términos **código** y **cifra** realmente tienen un significado ligeramente diferente, relacionado con las dos maneras en que pueden enviarse los mensajes. Cuando la información se transfiere letra por letra, se habla de cifrado. Si en cambio se transforman palabras completas o grupos de palabras en otras palabras o números utilizando un diccionario especial, a esto se le llama codificación. En la práctica, el término código se utiliza para referirse a códigos y cifras.

CODIGOS SECRETOS

La codificación y decodificación de mensajes secretos fue en otra época un área confinada principalmente a las actividades militares, o al menos a los Servicios de Inteligencia. Sin embargo hoy, el amplio uso de las líneas telefónicas públicas y de los canales de transmisión por satélite para transmisión de datos comerciales importantes, ha aumentado la necesidad de la codificación.

En los albores de la revolución informática, se utilizaban los comptómetros de **IBM** para romper los códigos de la Segunda Guerra Mundial. Desde entonces, todo nuevo avance en la tecnología de los ordenadores ha sido ansiosamente seguido por los maestros de espías y los





METODOLOGIA DE LA PROGRAMACION

El primer libro recomendado para escuelas de enseñanza de informática y para aquellas personas que quieren aprender la programación. Cubre las especificaciones del Ministerio de Educación y Ciencia para Estudios de Informática. Realizado por un alto mando del ejército Español, un Dr. Ingeniero y Diplomado en Informática y profesor de la UNED y por un oficial técnico especialista en informática de gestión. Utilizado en todos los institutos politécnicos del ejército español. Es un seguro candidado a ediciones en lengua inglesa, alemana y francesa. Es el primer libro que introduce a la lógica del ordenador. Es un elemento de base que sirve como introducción para la programación en cualquier otro lenguaje. No se requieren conocimientos de programación ni siquiera de informática. Abarca desde los métodos de programación clásicos a los más modernos. **Precio venta 2.200 ptas.**

MANUAL ESCOLAR PARA SU COMMODORE 64

Este libro, escrito especialmente para escolares de grado medio y superior, contiene muchos interesantes programas de aprendizaje para solucionar problemas, descritos detalladamente y de manera fácilmente comprensible. Facilitan un aprendizaje intensivo y ameno, con, entre otros, los siguientes temas: Teorema de pitágoras, progresiones geométricas, palanca mecánica, crecimiento exponencial, verbos irregulares, ecuaciones de segundo grado, movimientos de péndulo, formación de moléculas, aprendizaje de vocablos, cálculo de interés y su capitalización. Una corta repetición de los elementos BASIC más importantes y una introducción a los rasgos esenciales del análisis de problemas, entre otros, completan el conjunto. **Precio de venta 2.800 ptas.**

64 EN EL CAMPO DE LA TECNICA Y LA CIENCIA

Ofrece un campo fascinante y amplio de problemáticas científicas. Para esto el libro contiene muchos listados interesantes: Análisis de Fornier y síntesis, análisis de redes, exactitud de cálculo, formateado de números, cálculo del valor PH, sistemas de ecuaciones diferenciales, modelo ladrón presa, cálculo de probabilidad, medición de tiempo, integración, etc. **Precio venta 2.800 ptas.**

ROBOTICA PARA SU COMMODORE 64

En el libro de los robots se muestran las asombrosas posibilidades que ofrece el CBM 64, para el control y la programación, presentadas con numerosas ilustraciones e intuitivos ejemplos. El punto principal: Cómo puede construirse uno mismo un robot sin grandes gastos. Además, un resumen del desarrollo histórico del robot y una amplia introducción a los fundamentos cibernéticos. Gobierno del motor, el modelo de simulación, interruptor de pantalla, el Port-Usuario cómodo del modelo de simulación, Sensor de infrarrojos, concepto básico de un robot, realimentación unidad cibernética, Brazo prensor, Oir y ver. **Precio venta 2.800 ptas.**

TODO SOBRE EL FLOPPY 1541

La obra Standard del floppy 1541, todo sobre la programación en disquettes desde los principiantes a los profesionales, además de las informaciones fundamentales para el DOS, los comandos de sistema y mensajes de error, hay varios capítulos para la administración práctica de ficheros con el FLOPPY, amplio y documentado Listado del Dos. Además un filón de los más diversos programas y rutinas auxiliares, que hacen del libro una lectura obligada para los usuarios del Floppy. **Precio venta 3.200 ptas.**

MANTENIMIENTO Y REPARACION DEL FLOPPY 1541

Saberse apañar uno mismo, ahorra tiempo, molestias y dinero, precisamente problemas como el ajuste del floppy o reparaciones de la platina se pueden arreglar a menudo con medios sencillos. Instrucciones para eliminar la mayoría de perturbaciones, listas de piezas de recambio y una introducción a la mecánica y a la electrónica de la unidad de disco, hay también indicaciones exactas sobre herramientas y material de trabajo. Este libro hay que considerarlo en todos sus aspectos como efectivo y barato. **Precio venta 2.800**

EL MANUAL DEL CASSETTE

Un excelente libro, que le mostrará todas las posibilidades que le ofrece su grabadora de casettes. Describe detalladamente, y de forma comprensible, todo sobre el Datassette y la grabación en cassette. Con verdaderos programas fuera de serie: Autostart, Catálogo (¡busca y carga automáticamente!), backup de y a disco, SAVE de áreas de memoria, y lo más sorprendente: un nuevo sistema operativo de cassette con el 10-20 veces más rápido FastTape. Además otras indicaciones y programas de utilidad (ajuste de cabezales, altavoz de control). **190 pág. P.V.P. 1.600,- ptas.**

EL DICCIONARIO PARA SU COMMODORE 64

Este es el libro que buscaba: una enciclopedia exhaustiva del C 64 y su programación, un diccionario general de micros que contiene toda la terminología informática de la A a la Z y un diccionario técnico con traducciones de los términos ingleses de más importancia - los DICCIONARIOS DATA BECKER prácticamente son tres libros en uno. La increíble cantidad de información que contienen, no sólo los convierte en enciclopedias altamente competente, sino también en herramientas indispensables para el trabajo. El DICCIONARIO DATA BECKER se edita en versión especial para APPLE II, COMMODORE 64 e IBM PC. **350 pág. P.V.P. 2.800,- ptas.**

LENGUAJE MAQUINA PARA COMMODORE 64

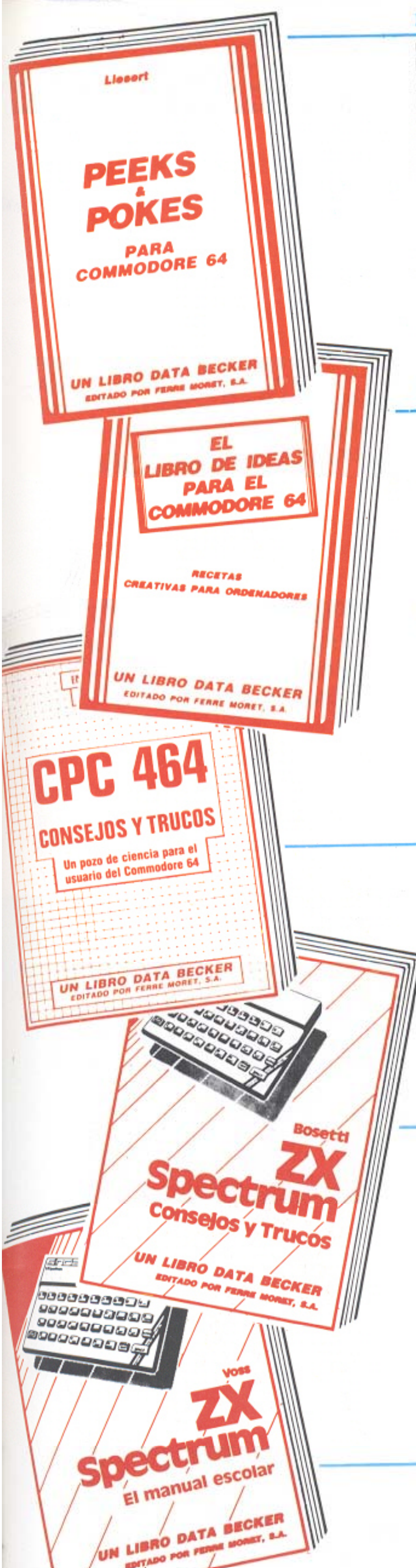
¡Por fin una introducción al código máquina fácilmente comprensible! Estructura y funcionamiento del procesador 6510, introducción y ejecución de programas en lenguaje máquina, manejo del ensamblador, y un simulador de paso a paso escrito en BASIC **1984, 201 pág. P.V.P. 2.200,- ptas.**

64 INTERNO

Con más de 60.000 ejemplares vendidos, ésta es la obra estándar para el COMMODORE 64. Todo sobre la tecnología, el sistema operativo y la programación avanzada del C-64. Con listado completo y exhaustivo de la ROM, circuitos originales documentados y muchos programas. ¡Conozca su C-64 a fondo! **1984, 352 pág. P.V.P. 3.800,- ptas.**

64 CONSEJOS Y TRUCOS

CONSEJOS Y TRUCOS, con más de 70.000 ejemplares vendidos en Alemania, es uno de los libros más vendidos de DATA BECKER. Es una colección muy interesante de ideas para la programación del Commodore 64, de POKEs y útiles rutinas e interesantes programas. Todos los programas en lenguaje máquina con programas cargadores en Basic. **1984, 364 pág. P.V.P. 2.800,- ptas.**





PEEKS y POKES

Con importantes comandos PEEK y POKE se pueden hacer también desde el Basic muchas cosas, para las que se necesitarían normalmente complejas rutinas en lenguaje máquina. Este libro explica de manera sencilla el manejo de PEEKs y POKEs. Con una enorme cantidad de POKEs importantes y su posible aplicación. Para ello se explica perfectamente la estructura del Commodore 64: Sistema operativo, interpretador, página cero, apuntadores y stacks, generador de caracteres, registros de sprites, programación de interfaces, desactivación del interrupt. Además una introducción al lenguaje máquina. Muchos programas ejemplo. **177 pág. P.V.P. 1.600,- ptas.**

MSX EL MANUAL ESCOLAR

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy completa y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cuadráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.
Precio de venta 2.800.

EL LIBRO DE IDEAS DEL COMMODORE 64

Casi todo lo que se puede hacer con el Commodore 64, está descrito detalladamente en este libro. Su lectura no es tan sólo tan apasionante como la de una novela, sino que contiene, además de listados de útiles programas, sobre todo muchas, muchas aplicaciones realizables en el C64. Se ha valorado especialmente, que el libro sea de fácil comprensión para los no iniciados. En parte hay listados de programas listos para ser tecleados, siempre que ha sido posible condensar «recetas» en una o dos páginas. Si hasta el momento no sabía que hacer con su Commodore 64, ¡después de leer este libro lo sabrá seguro! **1984, más de 200 páginas, P.V.P. 1.600,- ptas.**

MSX PROGRAMAS Y UTILIDADES

El libro contiene una amplia colección de importantes programas que abarcan, desde un desensamblador hasta un programa de clasificaciones deportivas. Juegos superemocionantes y aplicaciones completas. Los programas muestran además importantes consejos y trucos para la programación. Estos programas funcionan en todos los ordenadores MSX, así como en el SPECTROVIDEO 318 328. ESTRACTO DEL CONTENIDO: Volcado memoria hexadecimal. Editor gráficos. Editor de sonido. Escritura de ordenador. Lista referencia de variables. Calendario. Desensamblador. ADMINISTRACION de una colección de discos L.P. HOLLOW - JUEGO DE LAS CEREZAS. DIAGRAMAS DE BARRAS. TABLAS DEPORTIVAS. **1985, 194 pág. P.V.P. 2.200,- ptas.**

CPC-464 CONSEJOS Y TRUCOS

Ofrece una colección muy interesante de sugerencias, ideas y soluciones para la programación y utilización de su CPC-464: Desde la estructura del hardware, sistema de funcionamiento - Tokens Basic, dibujos con el joystick, aplicaciones de ventanas en pantalla y otros muchos interesantes programas como el procesamiento de datos, editor de sonidos, generador de caracteres, monitor de código máquina hasta listados de interesantes juegos. **Precio venta 2.200.**

MSX GRAFICOS Y SONIDOS

Las computadoras MSX no sólo ofrecen una relación precio/rendimiento sobresaliente, sino que también poseen unas cualidades gráficas y de sonido excepcionales. Este libro expone las posibilidades de los MSX de forma completa y fácil. El texto se completa con numerosos y útiles programas ejemplo. **Precio venta 2.800.**

ZX SPECTRUM CONSEJOS Y TRUCOS

Una interesante colección se sugestivas ideas y soluciones para la programación y utilización de su ZX ESPECTRUM. Aparte de muchos peeks, pokes y USRs hay también capítulos completos para, entre otros, entrada de datos asegurado sin bloqueo de ordenador, posibilidades de conexión y utilización de microdrives y lápices ópticos, programas para la representación de diagramas de barra y de tarta, el modo de utilizar óptimamente ROM y RAM.
Precio venta 2.200.

CPC-464 EL LIBRO DEL COLEGIO

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy compleja y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cudráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.
Precio venta 2.200.

ZX SPECTRUM EL MANUAL ESCOLAR

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy completa y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cudráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.
Precio venta 2.200.

TODOS LOS LIBROS, DISPONIBLES EN SONIMAG.

DATA BECKER

FERRE - MORET S.A.

TUSET, 8 ENTLO. 2.° - ☎ 218 42 04 - 218 40 58
TELEX 97851 CBCT E - 08006 - BARCELONA

BOLETIN DE PEDIDO
FERRE - MORET S.A.
Tuset n.° 8, entlo. 2.° Tel. 218 02 93
BARCELONA 08006
Deseo adquirir
Gastos envío: 300 ptas. ☐ Adjunto cheque ☐ Reembolso más gastos del mism[o]
NOMBRE
DIRECCION

reventadores de códigos. Actualmente, con una buena programación y la cifra correcta, un ordenador doméstico puede funcionar como cualquier máquina convencional de criptografía. Este es el primero de dos artículos que mostrarán cómo utilizar el ordenador para producir mensajes secretos codificados, siguiendo varios métodos diferentes que, como el propio espionaje, se sitúan a niveles diferentes de sofisticación.

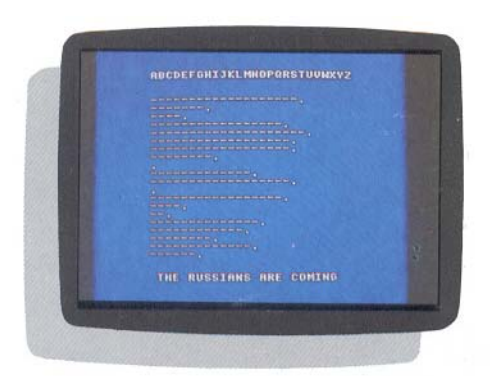
Aunque no seas un agente internacional, los métodos seguidos tienen interés por sí mismos, y siempre podrás servirte de ellos para enviar mensajes cifrados a otros amigos que tengan ordenador. De hecho, en este artículo hay un mensaje escondido en alguna parte.

CODIGOS DE DISTANCIA

La sucesión aparentemente aleatoria de símbolos mostrada hacia la derecha es de hecho un ejemplo de código de distancia. Como su nombre sugiere, se trata de un código que está basado en la distancia de un símbolo particular a un punto dado.

Este tipo de código fue utilizado hace más de 2000 años por el general griego **Lisandro.** Las distancias de las muescas a partir de la hebilla del cinturón de uno de sus esclavos componían un mensaje secreto que ayudó al general a derrotar al Imperio Persa.

Se puede formar sencillamente un código de distancia, poniendo las letras del alfabeto en la primera línea de un papel cuadriculado y componiendo el mensaje como se muestra en la pantalla. Mientras se ve la clave de las letras en la línea superior, el mensaje es fácil de entender. Sin embargo, si se suprime esta clave, no es fácil decodificar la información.



Pantalla del código de distancia.

Para hacer el descifrado todavía más intrincado, haz una rotación de la clave de las letras en la primera línea de la página. Se puede, por ejemplo, empezar por la N y cuando se llega a la Z, simplemente empezar otra vez por la A. A esto se le llama rotación cíclica.

El programa utiliza este método para producir una versión codificada del texto en claro, en el supuesto de que no se dejen espacios entre las palabras. Si realmente deseas enviar el mensaje, deberás tener acceso a una impresora, y asegurarte de que el destinatario del texto conoce el orden que ha de usar para la decodificación.

El programa funciona creando un bucle y sirviéndose de la facilidad de LEN para evaluar el valor equivalente ASCII de cada letra del texto en claro (líneas 90, 140). La mayoría de los ordenadores incorporan una función ASCII que permite representar las letras por medio de números. En todos ellos las letras toman los mismos valores. El uso de las funciones que manipulan cadenas de caracteres alfanuméricos, como LEN, vienen ampliamente descritas en los manuales de la máquina.

Después de convertir el mensaje en una serie de números, es fácil codificarlo utilizando una transformación lineal inmediata. Por ejemplo, en el programa, la letra V se traduce al equivalente ASCII de V menos 26 (línea 130).

Sólo falta utilizar la función TAB para imprimir el asterisco a la distancia adecuada a partir del lado derecho de la pantalla (línea 120) y el proceso de codificación está completo.

Teclea

```
30 PRINT"♥■>□■CODIGO DE
   DISTANCIA"
50 PRINT"■■NO DEJES
   ESPACIOS":PRINT"ENTRE
   PALABRAS!"
70 PRINT"■CUAL ES TU
   MENSAJE":INPUT A$
80 PRINT"♥"
90 FOR I=1 TO LEN(A$)
100 B$=MID$(A$,I,1)
110 V=ASC(B$)-45
120 IF V<=32 THEN PRINT
   TAB(V);"*":GOTO 150
130 V=V-26
140 PRINT TAB(V);"*"
150 NEXT I
```

UTILIZACION DEL CODIGO

Aunque el código de distancia pueda parecer demasiado sencillo para ser eficaz, tiene algunos factores a su favor. En primer lugar, antes de poder descifrar con éxito un código, hay que darse cuenta de que realmente éste existe. Y como es muy fácil enmascarar una sucesión de puntos (o asteriscos) aparentemente



aleatoria en un dibujo, por lo demás inocente, hay bastantes probabilidades de que un mensaje codificado de este tipo pase inadvertido.

Durante la última guerra mundial, los agentes alemanes utilizaron este truco. Una inspección detenida a la inocente imagen de un jardín, reveló que las pinzas de la ropa en una cuerda de tender componían un mensaje secreto.

Una manera de hacer el código de distancia aún más difícil de romper es reestructurar el programa de forma que la clave de las letras sea realmente aleatoria. Un experto consciente de que tiene ante sí un código de distancia necesita ensayar, como mucho, 26 combinaciones antes de resolver el problema. Sin embargo, si el orden de las letras de la clave es aleatorio, el número de combinaciones posibles aumenta de manera formidable.

EL CIFRADO DE ST. CYR

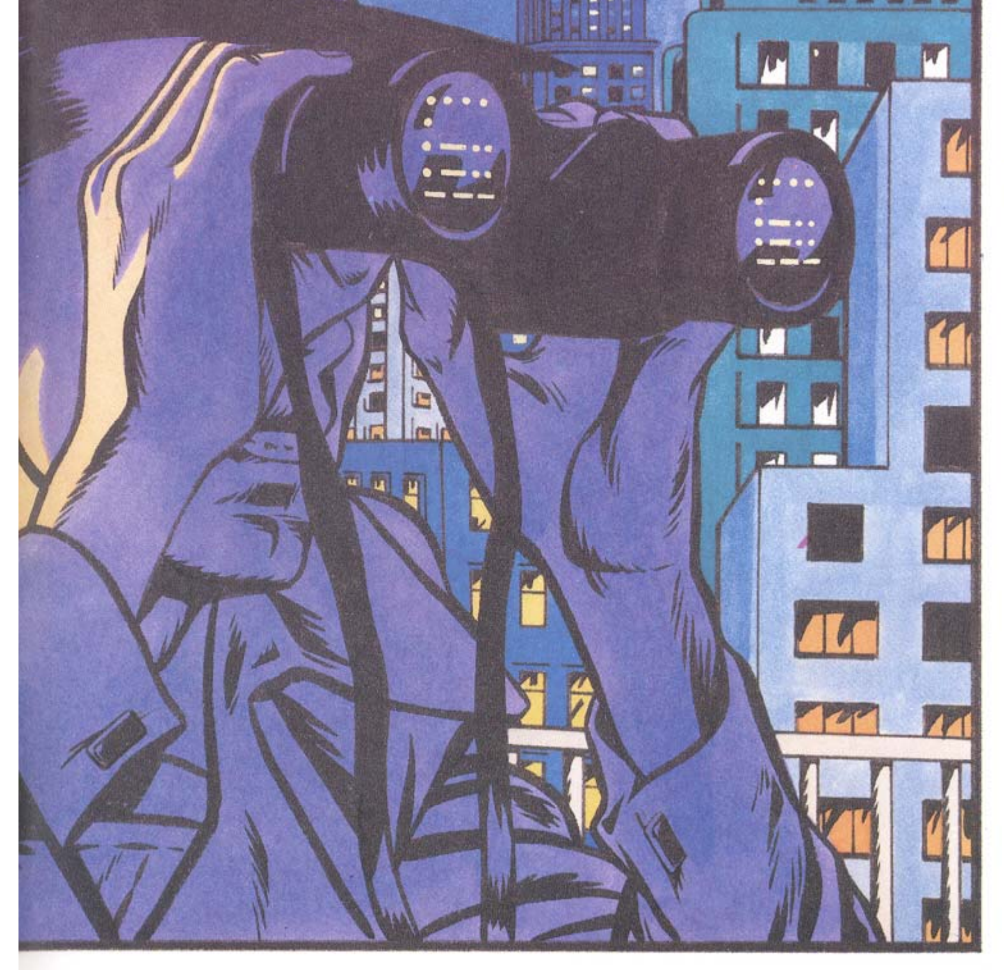
Los romanos tomaron el relevo de los griegos como maestros en criptografía. **Julio César** inventó un método de cifra por sustitución directa en el que cada letra se sustituía por la que estaba situada en el alfabeto tres lugares a la derecha. Así la A se convierte en D, la B se convierte en E, etc. Al final del alfabeto, la X se convierte en A, la Y en B y la Z en C. Utilizando este método el mensaje VIENE EL ENEMIGO se convierte en ZLHPH HN HKHOLJR.

Como se verá más adelante, el código de **César** es un caso especial del llamado cifrado de **St. Cyr,** y el resultado anterior se puede comprobar fácilmente ejecutando el siguiente programa y utilizando como número clave 3333333.

Con la caída del Imperio Romano cesaron los desarrollos en criptografía, y a pesar del creciente uso de códigos en los siglos XVI y XVII, hasta el siglo XIX no se produjeron mejoras significativas del código de **Julio César,** realizadas por la academia militar de **St. Cyr.** El cifrado de **St. Cyr** es sorprendentemente sencillo. Se compone con tres alfabetos en una escala deslizante. El alfabeto de abajo es el del texto en claro y sus equivalentes en cifra se toman de la línea superior. Partiendo del punto de origen mostrado en la figura, la palabra INPUT se transforma en AFHML. Una de las ventajas del cifrado de **St. Cyr** es que cada letra se puede codificar utilizando un alfabeto equivalente distinto. Esto hace muy difícil reventar un código.



*Pantalla del cifrado de **St. Cyr.***



En el programa de cifrado de **St. Cyr,** se ha incorporado esta utilidad por medio de un número clave, para obtener una seguridad extra. Todo el que tenga acceso a un listado de programa no podrá resolver el código, a menos naturalmente, que conozca también los siete números secretos.

Teclea

```
30 PRINT"♡■>□■CIFRA DE
   ST. CYR"
70 PRINT"■■(+1) CODIFICA
   EL MENSAJE"
80 PRINT"(-1) DECODIFICA EL
   MENSAJE"
90 INPUT"■■ESCRIBE 1 O
   -1■";S
100 PRINT"♡ESCRIBE TU
   MENSAJE":INPUT A$
```

```
120 PRINT"■ESCRIBE UNA
    CIFRA CLAVE DE":INPUT
    "SIETE CIFRAS■";N$
130 PRINT"□"
140 FOR K=1 TO LEN(A$)
145 M$=MID$(A$,K,1):IF
    M$<"A" OR M$>"Z" THEN
    PRINT M$;:GOTO 190
150 L=K-INT(K/7)*7+1
160 T=ASC(MID$(A$,K,1))+(S*
    VAL(MID$(N$,L,1)))
170 IF T<65 OR T>90 THEN
    T=T-(S*26)
180 PRINT CHR$(T);
190 NEXT K:PRINT
```

Este programa opera de una manera muy parecida al programa de códigos de distancia. Después de leer cada letra del texto en claro, se convierte primeramente en el equivalente numérico del carácter ASCII correspondiente. Luego se utiliza la función VAL para incrementar este número en una cantidad que depende del número deducido de su clave numérica (línea 160). Después de una comprobación para asegurarse de que el resultado está dentro de márgenes aceptables (línea 170), se aplica la utilidad CHR$ para sacar el mensaje codificado.

En los primeros días de junio de 1944 se podría haber codificado el siguiente mensaje: TROPASCAPTURADASENELPUENTEPEGASO. Utilizando el programa con el número clave 3821105, el texto codificado se convierte en: BTPQAXFIRUVRFGIUFOEQSCGOUEUHOCTP. Utilizando una variable indicadora S, que puede tomar los valores +1 ó −1, se puede utilizar el mismo programa para la decodificación.

El número clave de siete cifras depende de la elección personal. Sin embargo, como estas cifras se confían frecuentemente a la memoria, es una buena práctica recurrir a algún dato familiar, tal como algún número de teléfono.

Como hay diez millones de valores diferentes para el número clave, el cifrado de **St. Cyr** es realmente de decodificación muy laboriosa. No obstante, se puede introducir una dificultad adicional codificando el mismo mensaje dos veces, es decir, poniendo el mensaje ya codificado como entrada al programa codificador por segunda vez.

Considérese el mensaje MARTINEZ ES UN ESPIA. Y utilizando los números clave 2501000 y 2506900, la secuencia cifrado-descifrado es la siguiente:

Texto	Número clave	
MARTINEZESUNESPIA		
	2501000	Codifica
RASTINGEETUNEUUIB		
	2506900	Codifica
WAYCINIJEZDNEWZIH		
	2506900	Decodifica
RASTINGEETUNEUUIB		
	2501000	Decodifica
MARTINEZESUNESPIA		

CODIGO MORSE

Incluso el más seguro de los códigos es poco útil si no se puede transmitir rápidamente. **Napoleón** hizo algunos intentos para resolver el problema de la velocidad construyendo torres de señales por toda Francia. Para pasar mensajes de una torre a otra se utilizaba un semáforo primitivo.

Sin embargo, lo que realmente aceleró la transmisión de mensajes secretos fue la invención del telégrafo eléctrico y la cifra de sustitución conocida por código Morse. El inventor americano **Morse** estableció un código en que las letras estaban sustituidas por distintas combinaciones de puntos y rayas.

El programa del código Morse vale para cifrar texto, o decodificar una serie de puntos y rayas a sus letras adecuadas. Por ejemplo, la conocida señal de socorro SOS se convierte en /···/---/···/ y el texto ESCAPA A MEDIANOCHE pasa a ser /·/···/-·-·/·-/·--·/·-/·-/--/·/-··/··/·-/-·/---/-·-·/····/·/

Prueba los dos fragmentos de texto en el siguiente programa. Actuando en sentido inverso, si ejecutas el programa e introduces --0--0---000--0---***** aparecerá en la pantalla del televisor la palabra MAYDAY. Se utilizan los asteriscos para indicar al ordenador que se ha completado el mensaje.



En la primera parte de este programa se establece una cadena de caracteres con 130 puntos y rayas, para representar el equivalente Morse del alfabeto en orden secuencial (líneas 10-30). El signo menos es un buen símbolo para representar una raya, y hemos preferido el cero como símbolo gráfico para representar al punto, en lugar de utilizar el punto ortográfico.

Se ha previsto un campo de cinco caracteres para cada letra, cuando bastaría con cuatro para todo el alfabeto. Se ha hecho esto para dar la oportunidad de ampliar el programa a quien desee introducir números que tengan códigos más largos. La sección de codificación es muy parecida a la de los primeros programas. Cada letra del texto es leída y convertida en un número entre 1 y 26.

Al número equivalente se le aplica un factor de escala de cinco y la subcadena de puntos y rayas que resulta es lo que se imprime (línea 180).

Para la subrutina de descifrado que constituye la parte final del programa, se utiliza una técnica de búsqueda (líneas 230-250).

Después de leer las señales Morse, el ordenador buscará en la serie hasta que encuentre una subserie idéntica. Cuando esto suceda, es inmediato transformar la posición de la serie en el correspondiente código numérico ASCII. Nuevamente se utiliza la función CHR$ para sacar el resultado requerido.

Teclea

```
10 P$="Ø-□□□-ØØØ□-Ø-Ø□
   -ØØ□□Ø□□□□ØØ-Ø□--Ø
   □□ØØØØ□ØØ□□□"
20 P$=P$+"Ø---□-Ø-□□Ø-ØØ
   □--□□□-Ø□□□---□□Ø
   --Ø□--Ø-□Ø-Ø□□ØØØ□□"
30 P$=P$+"-□□□□ØØ-□□ØØØ
   -□Ø--□□-ØØ-□-Ø--□
   --ØØ□"
100 PRINT"▣■>□■CODIGO
   MORSE"
110 PRINT"■■TECLEA:"
115 PRINT"1 PARA CODIFICAR"
120 PRINT"2 PARA
   DECODIFICAR":INPUT N
130 PRINT"▣ESCRIBE TU
   MENSAJE":INPUT
   B$:PRINT"▣■";B$;"■=■"
140 IF N=2 THEN 190
150 FOR I=1 TO LEN(B$)
155 M$=MID$(B$,I,1):IF
   M$<"A" OR M$>"Z" THEN
   PRINT"/□□□□";:GOTO 185
160 K=ASC(MID$(B$,I,1))-64
170 T=1+(K-1)*5
180 PRINT"/"MID$(P$,T,5);
185 NEXT I:GOTO 280
190 FOR I=1 TO LEN(B$)
200 IF MID$(B$,I,1)<>"/"THEN
   NEXT I
210 D$=LEFT$(B$,I-1)+"□□□□
   ":B$=MID$(B$,I+1)
220 IF D$="*"THEN 280
230 FOR V=1 TO 26:W=1+(V-1)*5
240 IF LEFT$(D$,5)<>MID$(P$,
   W,5)THEN 260
250 PRINT CHR$(64+INT(W/5)
   +1);:GOTO 270
260 NEXT V:PRINT"□"
270 IF B$<>""THEN 190
280 PRINT
```

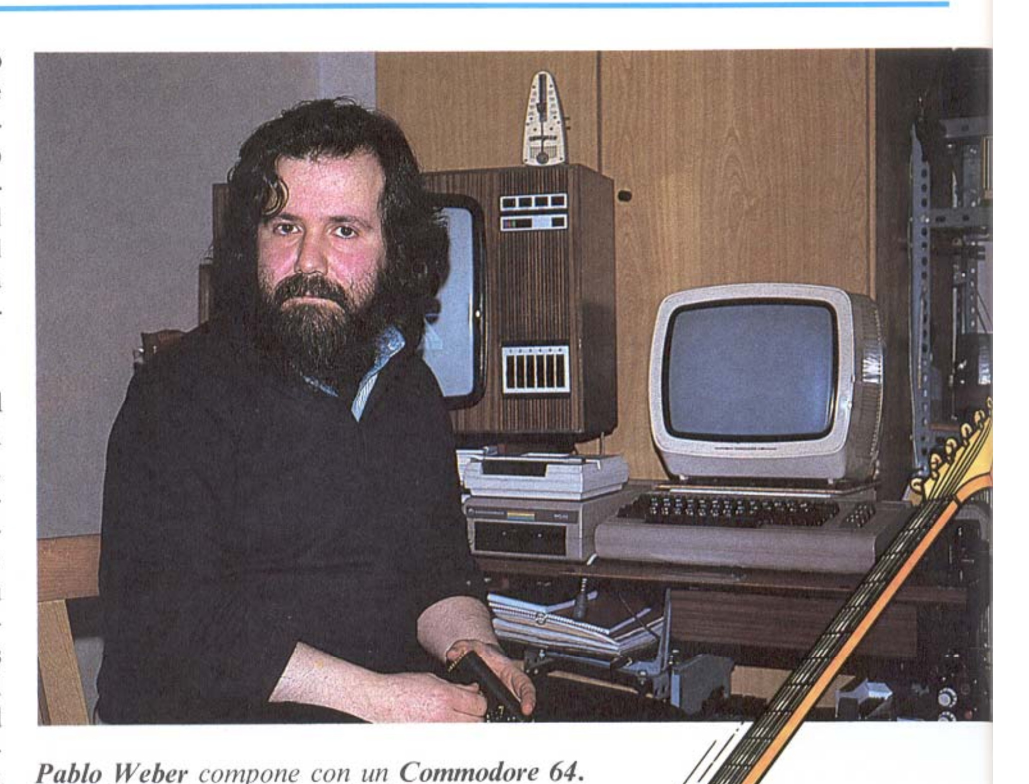
En un próximo artículo hablaremos de otros tipos de códigos.



UN ORDENADOR LOCO POR LA MUSICA

- EL ESTUDIO DE PABLO WEBER
- LOS PROGRAMAS DE RYO KAWASAKY
- COMPONER Y EJECUTAR MUSICA SIN ESTUDIAR SOLFEO

La compra de un ordenador abrió para él un horizonte nuevo, lleno de posibilidades desconocidas. Continuamente, había estado recopilando información procedente, en su mayoría, de revistas especializadas. Aquel cúmulo de datos dejaba entrever, al menos en teoría, lo que esconde la fría imagen de un modesto microordenador.



Pablo Weber** compone con un **Commodore 64.

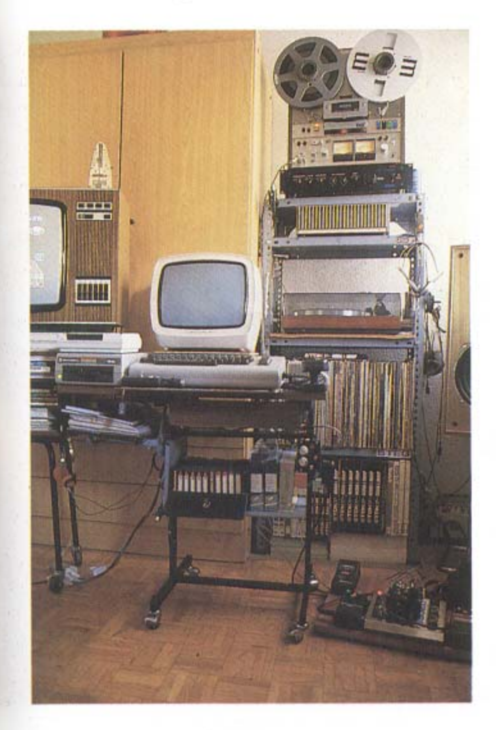


Pablo Weber, músico profesional que tiene en la guitarra eléctrica su principal instrumento de expresión, venía madurando la idea desde tiempo atrás. Pero, una vez ante el teclado, se enfrentó con la dura realidad: para sacarle el provecho soñado a su recién comprado **Commodore 64** había que tener sólidos conocimientos de programación o mejor aún, el *software* que glosaban con amplitud las revistas americanas. Surge la duda. ¿Será todo mentira o fruto de la calenturienta imaginación de los periodistas angloparlantes? Peregrinar por diversas tiendas especializadas de nuestro país no aportaba una luz nueva que diera solución a sus inquietudes. Los encargados no sabían darle cuenta de cómo hacerse con tales programas.

Poco a poco y con dificutlades fue haciéndose con el ansiado *software,* devorando virtualmente todo cuanto caía en sus manos. Esto le ha permitido hoy poder hacer comparaciones: «Cuando realmente me di cuenta de las posibilidades, se me hizo evidente la diferencia abismal que existe con los medios con los que venía trabajando, principalmente un magnetofón **Sony**.» A partir de entonces comprende que es posible acceder a cualquier tipo de composición, por compleja que sea. «Yo estoy trabajando mucho con métodos y sistemas musicales que tenía medio archivados» —afirma— «porque su estudio

es muy trabajoso. Pero los nuevos métodos con auxilio del ordenador facilitan enormemente el aprendizaje, y esto es aplicable a cualquier nivel. No le veo limitaciones».

El gran descubrimiento lleva un nombre propio: **Ryo Kawasaky,** un



Su estudio es pequeño pero completo.

músico de origen japonés afincado en los Estados Unidos desde mediados de los setenta. Es un enamorado de la música y el sonido. Su historia presenta cierto paralelismo con la de **Pablo Weber.** Hace unos tres años decidió comprarse también un **Commodore 64,** y desde entonces, la mayor parte de sus desvelos han sido copados por el artilugio informático. Fruto de este trabajo son dos paquetes de gran éxito comercial en el mercado americano que llevan por títulos: «Kawasaky Synthesizer» y «Kawasaky Rhythm Rocker». Los conocimientos acopiados durante los estudios de licenciatura en ciencias físicas fueron la sólida base que permitió al músico japonés familiarizarse rápidamente con los pormenores de la programación en código máquina. Su dominio de los infinitos recursos del versátil *chip* SID es asombroso, a juzgar por los resultados.

El «Kawasaky Synthesizer» se divide en dos partes. Una está orientada a la ejecución de piezas musicales y la otra a la composición.

UN SONIDO ESPECTACULAR

En su estudio, **Pablo Weber** nos hace esperar los obligados minutos necesarios para la carga del programa desde el *diskette* al ordenador y a continuación lleva a cabo las selecciones correspondientes en el teclado. El resultado audible es realmente sorprendente. Apenas podemos creer que el **Commodore 64** sea quien produce esas notas. En realidad cabe aclarar que **Weber** mejora sustancialmente el sonido resultante. En la salida prevista para el audio —en la parte posterior del ordenador— ha conectado una cadena de dispositivos electrónicos, habitualmente utilizados por él con la guitarra, «tampoco se trata de equipo caro y complejo». Un retardo largo, de naturaleza analógica, produce un efecto de cámara (reververación, eco). La unidad de retardo corto produce una nueva señal idéntica a la original, pero retrasada unos microsegundos. Un octavador que es otro de los elementos auxiliares, produce formas de onda una o dos octavas por debajo de la frecuencia fundamental. El punto final de la cadena lo componen un amplificador de audio y un par de *baffles*.

Una vez escuchado parte del trabajo elaborado con el sintetizador del **Sr. Kawasaky,** tenemos oportunidad de comprobar cómo puede componer un músico ayudado por la segunda parte del programa. El contenido de la pantalla no puede ser más simplificado, para un usuario no informático. Las tres rejillas que visualiza el televisor (o monitor) representan cada una a las correspondientes voces del **64.** Las notas se depositan en ellas directamente desde el teclado. La presentación de este paquete guarda cierto parecido con otro algo más conocido en nuestro mercado, el «MusiCalc».

A continuación, **Weber** nos muestra el «Music Construction Set», otro versátil paquete capaz de llevarnos a pensar que cualquier analfabeto mu-

*Pantalla de uno de los programas de **Kawasaky:** el «Rhythm Rocker».*





Las unidades de retardo de su guitarra son las únicas sofisticaciones.

sical puede ejecutar una melodía simplemente copiando en la pantalla el pentagrama al pie de la letra. La pantalla de trabajo, esta vez más familiar para quien tiene conocimientos musicales, se divide en dos áreas horizontalmente: arriba dos pentagramas en blanco esperan ser llenados con notas; abajo se dispone de la mayoría de los símbolos empleados en la nomenclatura musical (la duración de las notas, las claves, el tirmo, etc.) y una serie de casillas indicadoras de las posibilidades de manipulación musical. Todo el control se ejerce mediante una mano con el dedo índice estirado. Obligando al dedo a señalar una duración, por ejemplo una corchea, y seguidamente un lugar en uno de los dos pentagramas, la nota queda dispuesta de igual manera que si hubiera sido dibujado sobre el papel, pero con la ventaja de poder ejecutarla en cualquier momento. **Weber** nos comenta: «La aventura no es el sonido que se pueda obtener con el ordenador, sino la facilidad de ejecución de instrumentos que a lo mejor ni siquiera conoces en la práctica, pero que tienes la idea en mente de cómo componer para ellos. Todo lo que tengas en mente, dependiendo de si eres creativo y te acompaña el talento, es posible plasmarlo. El trabajo no te lo va a quitar nadie, pero accedes directamente al sonido, que en música es fundamental.» Un ejemplo nos aclara las ideas. Puede plantearse la cuestión de elaborar una secuencia rítmica, donde hay un contrabajo y una batería. Según **Weber,** «Ese planteamiento es una creación para terceros, porque yo no sé tocar el contrabajo y la batería, pero esto es lo que me gustaría escuchar. La solución pasa por un trabajo previo con la partitura, al que sigue el de comunicar a los músicos correspondientes la idea. Aquí es donde pueden surgir una serie de valoraciones y matices. Es muy lógico que cada cual tenga su interpretación». La frustración de compositor deja de tener sentido con ayuda del ordenador, «eres capaz de explicar exactamente lo que deseas, pues aunque en música hay unos signos muy concretos, otros no lo son tanto y se prestan un poco a la subjetividad del intérprete».

UNA FORMA DE POTENCIAR LA CREATIVIDAD

Sin embargo, no todo son elogios para el chip SID del **64.** A pesar de sus brillantes características, **Weber** reconoce que también tiene importantes limitaciones, aunque a grandes rasgos sea capaz de ofrecer prestaciones similares a las de otros sintetizadores especializados.

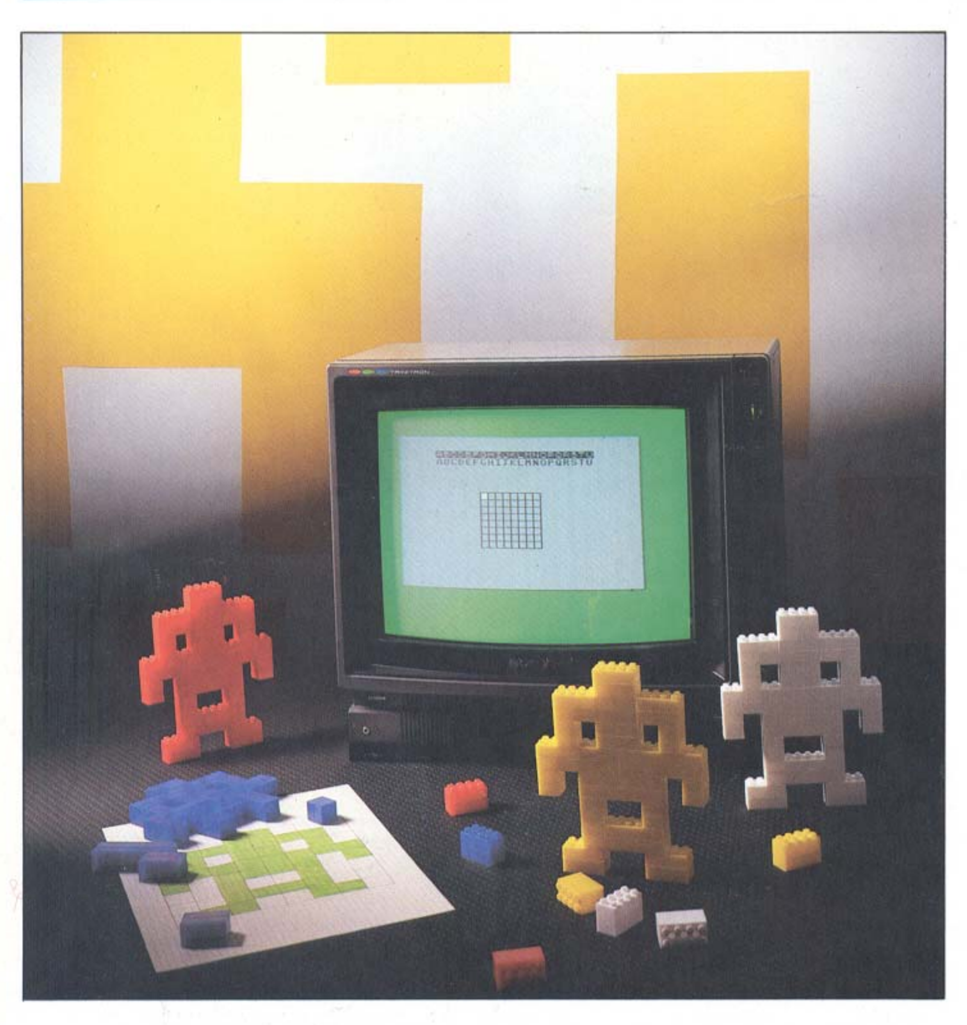
Para este joven músico la creatividad sigue siendo el caballo de batalla. «Para hacer algo interesante con un sintetizador o un ordenador es necesario tener el mismo grado de conocimientos que para hacerlo con una flauta, o unas maracas» —añade en tono de broma— «quien tiene ideas obtiene resultados. En el caso contrario, tal vez llegaría el momento en que no supiera dar una simple nota». Desde luego más que límites a la creatividad, ve todo lo contrario: la magnífica posibilidad de potenciarla.

Otra notable ventaja que ha encontrado a su nuevo instrumento es el ahorro de tiempo con el material de estudio: «a lo mejor tenías que tirarte semanas con él para llegar a conclusiones. Ahora necesito días para llegar a las mismas conclusiones con menos desgaste de energía y comprendiéndolo mejor».

Todavía queda mucho camino por recorrer. Al parecer, los músicos siguen siendo reacios a estas modernidades. **Pablo Weber** sigue trabajando con entusiasmo y desea contactar con cuantos tengan inquietudes en este sentido. De cualquier modo INPUT puede hacer llegar a **Pablo Weber** las cartas que lleguen a la redacción procedentes de sus lectores.

DISEÑA TUS CARACTERES GRAFICOS

- ALMACENAMIENTO DE LOS GDU EN UN BANCO DE MEMORIA
- GRABACION DE LOS GDU EN CINTA O DISCO
- TECLAS DE CONTROL



Si estás cansado del trabajo laborioso que supone la creación de los GDU (Gráficos Definidos por el Usuario) sobre un papel, teclea este programa y tómate un descanso mientras la máquina realiza el trabajo. A continuación podrás dibujar las formas que desees: desde un carácter chino hasta lo último en marcianitos de cuatro patas.

Los GDU son una de las herramientas más versátiles utilizadas por los programadores de gráficos. Pero uno de los inconvenientes que conllevan, si quieres diseñar tus propios dibujos, es el esfuerzo que entrañan: En primer lugar hay que dibujarlos sobre el papel, luego elaborar las sentencias DATA y por último POKEar dentro de la memoria; si además has cometido algún error, te tomará bastante tiempo ver los resultados. Sólo tras la realización de este proceso es posible visionar los GDU en la pantalla de tu monitor.

Por ello presentamos aquí un programa muy distinto a todo lo anteriormente expuesto. Por un lado reemplaza a la hoja de papel cuadriculado por una cuidada presentación en la pantalla, que te permite dibujar cuadraditos (*pixels*) donde desees. En lugar de tener que trabajar a mano dando valores a las sentencias DATA, éstos se obtienen de modo automático.

Pero el valor real del programa radica en que también te permite ver el GDU tal y como está construido —a tamaño natural— evitando la molestia de introducir las DATA dentro de otro programa para esta comprobación. Varias opciones de control permiten reajustar los GDU con la simple presión de una tecla —quizás te gustaría verlo en inverso, o girarlo lateralmente de izquierda a derecha.

Una vez realizado el GDU imaginado desearás que sea posible conservarlo —no olvidemos que dispones de 20 millones de millones de millones de opciones—. De este modo el programa te permite almacenar un banco de GDU terminados al que puedes recurrir cuantas veces precises. También puedes guardar (SAVE) los GDU, sacándolos del programa para tenerlos en una cinta, y cargarlos (LOAD) luego, dentro de cualquier programa, para utilizarlos en la pantalla —u otra vez en el generador de GDU para una edición posterior.

El programa consta de dos partes. La primera parte del listado, que aquí se muestra, nos permite crear el generador básico de GDU. Proporciona una rejilla en el centro de la pantalla, permitiendo movernos por toda ella, para dibujar el patrón de *pixels* que deseamos haya en el GDU final. En la siguiente parte del artículo verás las rutinas que permiten modificar los GDU en aspectos más sutiles, así como ser visualizados en la pantalla.

Te puedes ahorrar tener que teclearlo de nuevo la próxima vez, grabándolo (SAVE) en cinta cuando esté completamente copiado de la revista.

Los programas que se dan a continuación son para **Commodore 64** y **Vic-20.** Ambos utilizan las mismas teclas de control y su funcionamiento es similar. El programa para **Vic-20** es para un **Vic** sin ampliación de memoria o para una máquina que disponga de 3K de expansión. No funcionará en los **Vic-20** ampliados a 8K ó 16K —no obstante siempre es posible prescindir del paquete de ampliación.

Cuando introduzcas y ejecutes (RUN) los programas en los **Commodore,** existirá un retraso de aproximadamente un minuto mientras el ordenador limpia el espacio para los nuevos caracteres y POKEa el conjunto existente en la memoria RAM.

Tan pronto como esto haya sido realizado, la pantalla mostrará cambios, proporcionando la rejilla en la que puedes definir tus caracteres.

Puedes mover el cursor por toda la rejilla utilizando las teclas: "Z" para ir hacia la izquierda; "X" para la derecha; ";" para arriba; y "/" para abajo. Cuando el cursor está sobre el *pixel* que deseas activar, presiona la tecla SHIFT. Puedes rellenar un conjunto de *pixels* rápidamente presionando SHIFT LOCK y moviéndolo simultáneamente.

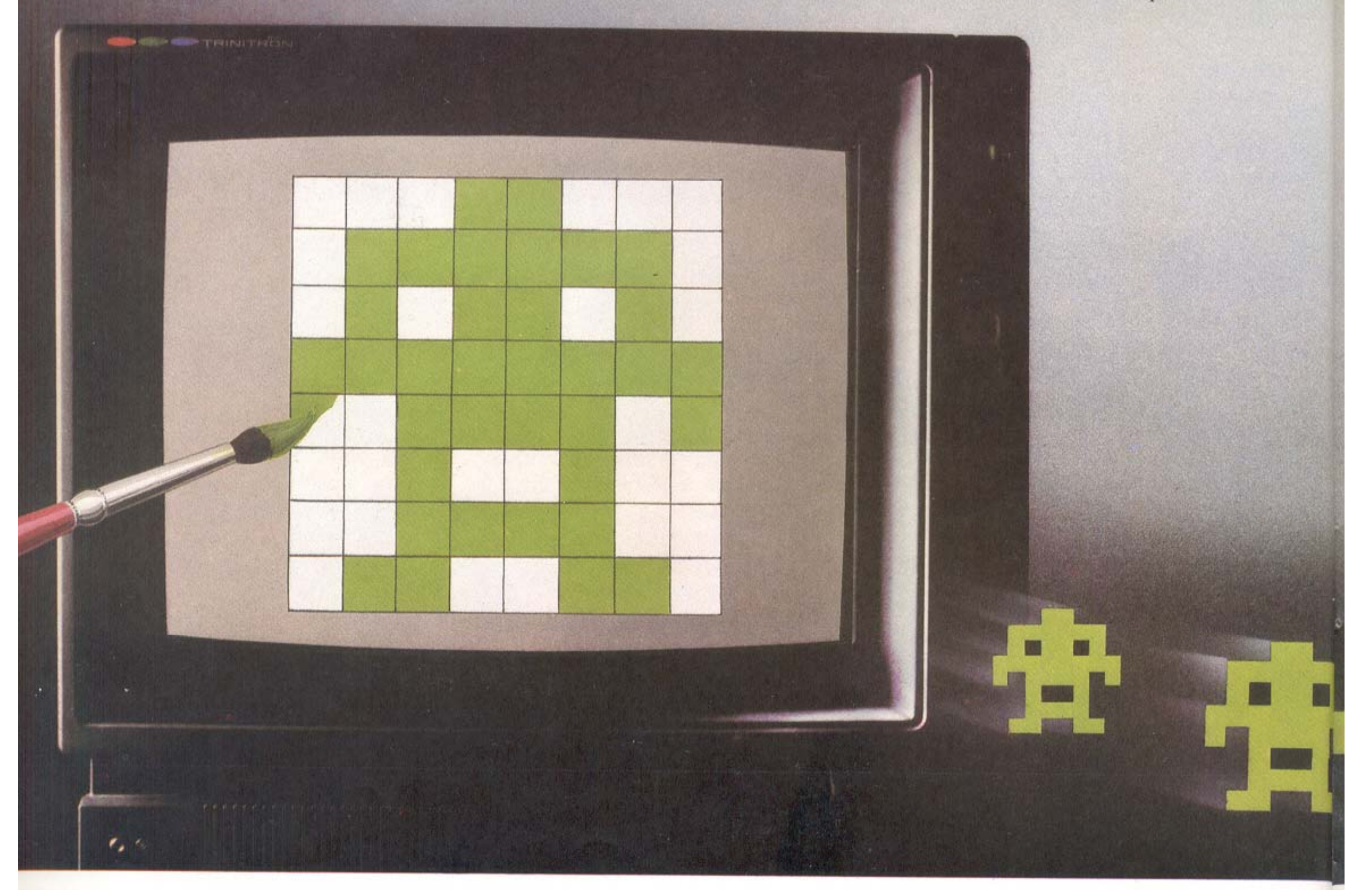
Tan pronto como comiences a activar *pixels,* podrás desactivarlos de nuevo presionando la tecla Commodore.

Mediante el movimiento por el interior de esta rejilla, activando los *pixels* que desees, puedes crear muy rápidamente tus GDUs; además el programa ayuda proporcionándote un conjunto de ventajas adicionales. La primera de ellas es contar con una línea idéntica de caracteres bajo el cuadrado. Esta muestra el banco de memoria actual de los GDU así como sus posiciones. En la línea superior de la línea están simplemente las letras de la A a la Z, mientras que en la inferior aparece una serie de GDUs que comienzan como los caracteres normales, pero pueden ser reemplazazados por tus GDUs propios. Este banco de almacenamiento es utilizado cuando llega el momento de almacenar sus caracteres.

COMO ALMACENAR UN GDU

Una vez satisfecho con el diseño del GDU, presiona la tecla de la flecha hacia la izquierda. El ordenador preguntará bajo qué letra deseas almacenar el GDU. Comprueba lo que visualiza el banco de memoria elegido para ver qué carácter quieres reemplazar por el recién definido GDU.

Cuando hayas decidido dónde situarlo, presiona la tecla correspondiente a la letra deseada. Después de una breve pausa el carácter ocupará su lugar en el banco de memoria, en la línea inferior de la pantalla.



Una vez que hayas diseñado y almacenado un carácter en el banco, puedes recuperarlo para su edición en la rejilla, en cualquier momento, presionando la tecla @, seguida del carácter deseado.

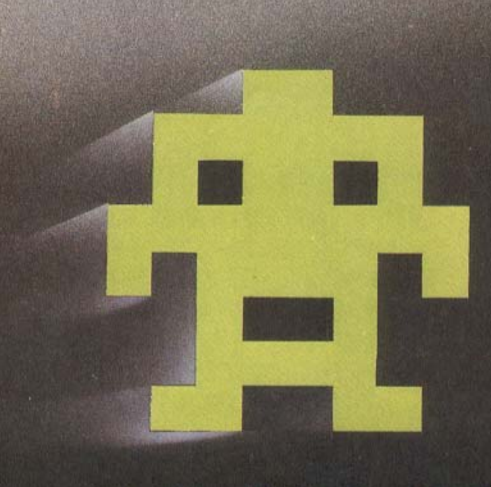
Existe un carácter más de control en esta mitad del programa, la tecla "£". Esta proporciona el acceso al modo de cinta, para poder grabar, bien (SAVE) el actual banco de caracteres, o bien, cargar (LOAD) lo que previamente hayamos guardado.

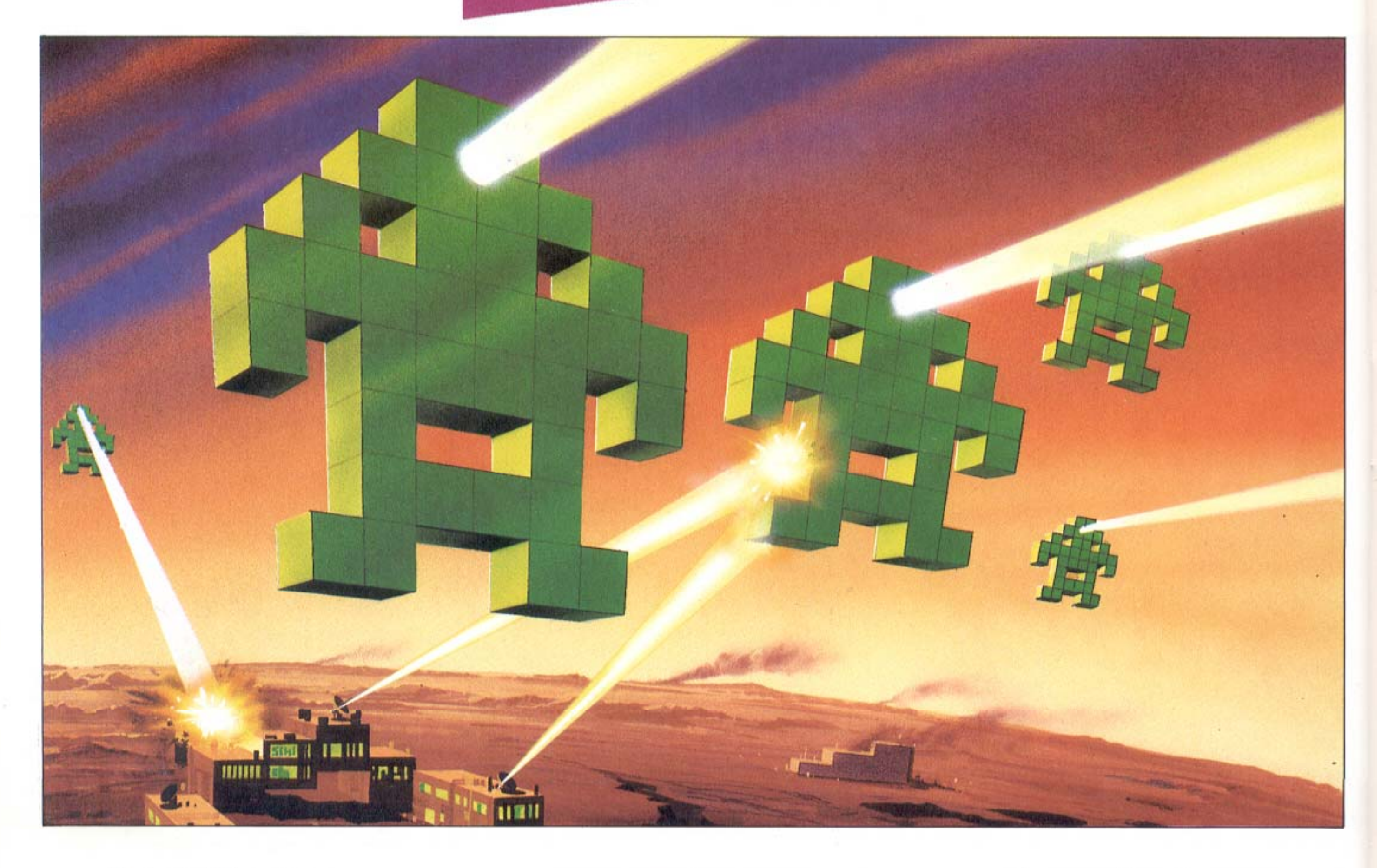
Tras presionar esta tecla, el ordenador te pregunta si quieres grabar (SAVE) o cargar (LOAD). Se supone que la cinta se encuentra preparada en el lugar correcto y lista para iniciar la carga o grabación. Cuando el ordenador ha finalizado su trabajo te devuelve la pantalla de edición para que continúes.

El próximo artículo de esta serie completará el programa y utilizará unas teclas de control más para rotar, efecto espejo, invertir, limpiar la rejilla e imprimir las DATA en una impresora. También habrá instrucciones para reclamar a los GDU y utilizarlos dentro de otro programa.

Teclea para Vic-20

```
10 PRINT CHR$(8):POKE
   51,255:POKE 52,27:POKE
   55,255:POKE 56,27:CLR
20 FOR Z=0 TO 511:POKE
   7168+Z,PEEK(32768+Z):
   NEXT Z
30 PRINT"■":POKE
   36879,25:POKE
   36869,255
33 FOR Z=0 TO 7:POKE
   7168+Z,255:NEXT Z
35 PRINT"■■GENERADOR
   DE":PRINT"□□CARACTERES"
38 PRINT"■■□□□□□□□□□
   □□□□□□□□□□□"
40 A$="ABCDEFGHIJKLMNOPQRS
   TUV"
50 X=0:Y=0:Z(0)=128:Z(1)=6
   4:Z(2)=32:Z(3)=16:Z(4)=
   8:Z(5)=4:Z(6)=2:Z(7)=1
100 PRINT"■■■■■■■■■■■
    □□12345678"
110 FOR Z=1 TO 8:PRINT A;"■■
    □□□□□□□□□":NEXT
120 PRINT"■□□□□□□□□□
    □□":PRINT"■"A$:
    PRINT A$
130 XX=X:YY=Y:C=7880+X+Y*22
    :CC=PEEK(C)
140 POKE
    C,233:P=PEEK(197):POKE
    198,0
150 IF P=33 AND X>0 THEN
    X=X-1
155 IF P=26 AND X<7 THEN
    X=X+1
160 IF P=22 AND Y>0 THEN
    Y=Y-1
165 IF P=30 AND Y<7 THEN
    Y=Y+1
170 IF PEEK(653)=1 THEN
    POKE C,230:GOTO 130
180 IF PEEK(653)=2 THEN
    POKE C,207:GOTO 130
200 POKE C,CC
250 IF P=8 THEN 1200
280 IF P=53 THEN 1600
290 IF P=6 THEN 1400
300 GOTO 130
1200 GOSUB 1500
1235 FOR Z=0 TO 7:T=0:FOR
     ZZ=0 TO 7
1240 C1=7880+ZZ+Z*22:C2=PEEK
     (C1)
1250 IF C2=230 THEN
     T=T+Z(ZZ)
1260 NEXT ZZ:POKE
     7168+(A*8)+Z,T:POKE
     7168+Z,T
1280 NEXT Z:GOTO 130
1400 POKE 198,0:POKE
     36869,240:N$="":INPUT"■
     ESCRIBE EL
     NOMBRE";N$:IF
     N$=""THEN 30
1405 PRINT"■(S)AVE O
     (L)OAD?":OU=0
1406 GET K$:IF K$="S" THEN
     OU=1:GOTO 1410
1407 IF K$<>"L" THEN 1406
1410 PRINT"■":OPEN
     1,1,OU,N$:REM**PARA
     DISCO:OPEN
     1,8,OU,"@:"+N$
1413 FOR Z=7176 TO 7352:IF
     OU=1 THEN
     PRINT#1,PEEK(Z):GOTO
     1420
1415 INPUT#1,ZZ:POKE Z,ZZ
1420 NEXT Z:CLOSE 1:GOTO 30
1500 PRINT"■■■■■□□□□□
     ESCRIBE LA LETRA":FOR
```



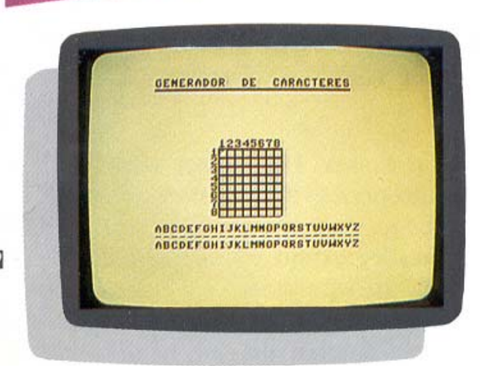


```
      Z=1 TO 25:NEXT Z
1505 PRINT"■■■■■□□□□□
      □□□□□□□□□□□□□□□
      □□"
1510 GET A$:IF A$<"A" OR
      A$>"V" THEN 1500
1520 A=ASC(A$)−64:RETURN
1600 GOSUB 1500
1610 FOR ZZ=0 TO 7
1611 AA=PEEK(7168+A*8+ZZ)
1612 FOR Z1=0 TO 7
1614 IF AA-Z(Z1)=>0 THEN
      AA=AA-Z(Z1):POKE
      7880+Z1+ZZ*22,230:GOTO
      1620
1616 POKE 7880+Z1+ZZ*22,207
1620 NEXT Z1,ZZ:GOTO 130
```

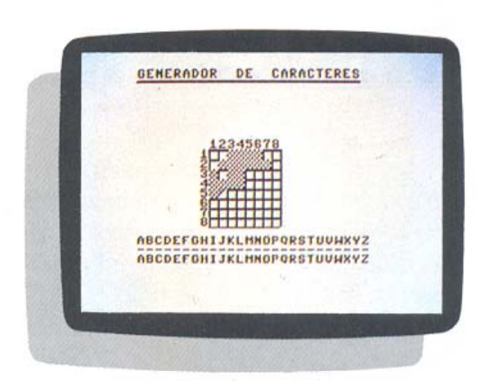
Teclea para Commodore-64

```
   5 PRINT"  ESPERA UN
     MINUTO"
  10 POKE 51,255:POKE
     52,47:POKE 55,255:POKE
     56,47:CLR
  15 POKE 56334,0:POKE 1,51
  20 FOR Z=0 TO 1023:POKE
     12288+Z,PEEK(53248+Z):
     POKE 13312+Z,PEEK
     (53248+Z)
  22 NEXT Z
  25 POKE 1,55:POKE 56334,1
  30 PRINT"■■";CHR$(8):POKE
     53280,1:POKE 53281,1:
     POKE 53272,28
  34 PRINT TAB(7);"■
     GENERADOR□□DE□□
     CARACTERES"
  38 PRINT TAB(7);"□□□□□□
     □□□□□□□□□□□□□□
     □□□"
  40 A$="ABCDEFGHIJKLMNOPQRS
     TUVWXYZ"
  50 X=0:Y=0:Z(0)=128:Z(1)=6
     4:Z(2)=32:Z(3)=16:Z(4)=
     8:Z(5)=4:Z(6)=2:Z(7)=1
 100 PRINT"■■■■■■■■■
     ■";TAB(15);"12345678"
 110 FOR Z=1 TO 8:PRINT
     TAB(13);Z"■■□□□□□□
     □□":NEXT
 120 PRINT TAB(15);"■□□□
     □□□□□":PRINT"■";
     TAB(7);A$:
     PRINT TAB(7);
 122 FOR Z=1 TO
     26:PRINT"■-";:NEXT
     Z:PRINT:PRINT TAB(7);A$
 130 XX=X:YY=Y:C=1399+X+Y*40
     :CC=PEEK(C)
 140 POKE C,233:P=PEEK(197):
     POKE 198,0
 150 IF P=12 AND X>0 THEN
     X=X−1
 155 IF P=23 AND X<7 THEN
     X=X+1
 160 IF P=50 AND Y>0 THEN
     Y=Y−1
 165 IF P=55 AND Y<7 THEN
     Y=Y+1
 170 IF PEEK(653)=1 THEN
     POKE C,230:GOTO 130
 180 IF PEEK(653)=2 THEN
     POKE C,207:GOTO 130
 200 POKE C,CC
 250 IF P=57 THEN 1200
 280 IF P=46 THEN 1600
 290 IF P=48 THEN 1400
 300 GOTO 130
1200 GOSUB 1500
1235 FOR Z=0 TO 7:T=0:FOR
     ZZ=0 TO 7
1240 C1=1399+ZZ+Z*40:C2=PEEK
     (C1)
```

```
1250 IF C2=230 THEN
     T=T+Z(ZZ)
1260 NEXT ZZ:POKE
     12288+(A*8)+Z,T:POKE
     12288+Z,T
1280 NEXT Z:GOTO 130
1400 POKE 198,0:POKE
     53272,21:N$="":INPUT"▢
     ESCRIBE EL NOMBRE";N$:
     IF N$=""THEN 30
1405 PRINT"▣(S)AVE O
     (L)OAD?":OU=0
1406 GET K$:IF K$="S" THEN
     OU=1:GOTO 1410
1407 IF K$<>"L" THEN 1406
1410 PRINT"▢":R$=CHR$(13):
     OPEN 1,1,OU,"@:"+N$
1413 FOR Z=12296 TO 12504:IF
     OU=1 THEN
     Z$=STR$(PEEK(Z)):Z$=
     RIGHT$(Z$,LEN(Z$)-1)
1414 IF OU=1 THEN
     PRINT#1,Z$R$:GOTO 1420
1415 INPUT#1,Z$:POKE
     Z,VAL(Z$)
1420 NEXT Z:CLOSE 1:GOTO 30
1500 PRINT"▣▣▣▣▣";TAB(14)
     ;"ESCRIBE LA LETRA":FOR
     Z=1 TO 25:NEXT Z
1505 PRINT"▣▣▣▣▣";TAB(14)
     ;"□□□□□□□□□□□□□□
     □□□"
1510 GET A$:IF A$<"A" OR
     A$>"Z" THEN 1500
1520 A=ASC(A$)-64:RETURN
1600 GOSUB 1500
1610 FOR ZZ=0 TO 7
1611 AA=PEEK(12288+A*8+ZZ)
1612 FOR Z1=0 TO 7
1614 IF AA-Z(Z1)=>0 THEN
     AA=AA-Z(Z1):POKE
     1399+Z1+ZZ*40,230:GOTO
     1620
1616 POKE 1399+Z1+ZZ*40,207
1620 NEXT Z1,ZZ:GOTO 130
```



Para la realización de los GDU el programa pone a nuestra disposición una parrilla de 8×8 pixels.



Una vez terminado el diseño del GDU podemos guardarlo en memoria y pasarlo después a una cinta.

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.

b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

Envía tu mensaje a:

INPUT COMMODORE-ZOCO
c./ Alberto Alcocer, 46
28016 **MADRID**

MISION IMPOSIBLE



Nuestro personaje se encuentra prisionero, tratando de encontrar salida por medio del ascensor.

«Another visitor, stay well, stay here forever», dice una amenazadora voz al principio del programa. Acabas de entrar en una galería, «Permanecerás aquí para siempre».

Penetraste, por error, en la madriguera del maléfico doctor **KNOT.** Un auténtico laberinto con más de 50 salas, que el desarrollo del juego se encarga de repartir aleatoriamente.

Desconecta el ordenador, o simplemente teclea NEW RETURN antes de introducir este programa:

Apareces en uno de los ascensores que te llevan arriba y abajo por las distintas pantallas que conforman la compleja instalación.

El malvado doctor ha tenido mucho cuidado de esconder los planos que te permitirán salir de este mal sueño. Y, sin embargo, debes encontrarlo; tu vida depende de ello. Pasadas seis horas serás electrocutado si no has logrado hallar la salida.

Una vez que has caído en tan perfecta trampa, empieza tu misión imposible: el doctor está dispuesto a entorpecer en todo lo posible tu huida y no duda en poner los medios más sofisticados y refinados para que llegues a pensar que no hay posibilidades de escapar.

En cada pantalla te perseguirán inteligentes robots, que detectan con increíble rapidez tu presencia y ubicación exacta. Has de ser muy hábil y cauto para evitarlos, porque disponen de un rayo mortífero presto a ser disparado. Como única arma dispones de tu asombrosa facilidad para dar saltos mortales (es uno de los mejores detalles del juego) y correr hacia delante.

En otras pantallas aparecerá una enorme bola, que te perseguirá con la poco sana intención de deshacerse de ti desintegrándote.

¿Cómo poder escapar? En cada sala vas a tener que «rebuscar» en los muebles para encontrar los planos que te salvarán la vida. Los robots que las habitan tienen una misión concreta, que les obliga a proteger los muebles que son tu esperanza.

Horizontalmente la pantalla se divide en dos. En la parte superior, ves la habitación donde estás situado, mientras que la ventana inferior muestra un mapa en blanco de la madriguera. El mapa se va desarrollando ante tus intrigados ojos a medida que recorrer las sucesivas habitaciones/pantalla o avanza el ascensor. Esta parte te sirve también para ir confeccionando el plano total con los mapas que has encontrado en los muebles.

No te preocupes, sólo tienes que organizar un rompecabezas de cincuenta piezas. El ordenador te ofrece su ayuda, disponiéndolas en los distintos sentidos posibles para obtener una interpretación coherente. Incluso puedes preguntarle si las vas montando bien.

Estamos sin duda ante uno de los mejores juegos existentes en el mercado del **C-64,** tanto en lo referente a las presentaciones gráficas como a la de la síntesis de voz, aunque solamente hable en inglés (de un realismo sorprendente..., ¿quién se ha escondido dentro del monitor?).

DATOS GENERALES	CALIFICACION
TITULO: Mission Imposible	GRAFICOS: 5 sobre 5
FABRICANTE: Epyx	COLOR: 4,5 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 4 sobre 5
MEMORIA: 64 K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Espionaje	REALISMO: 4,5 sobre 5

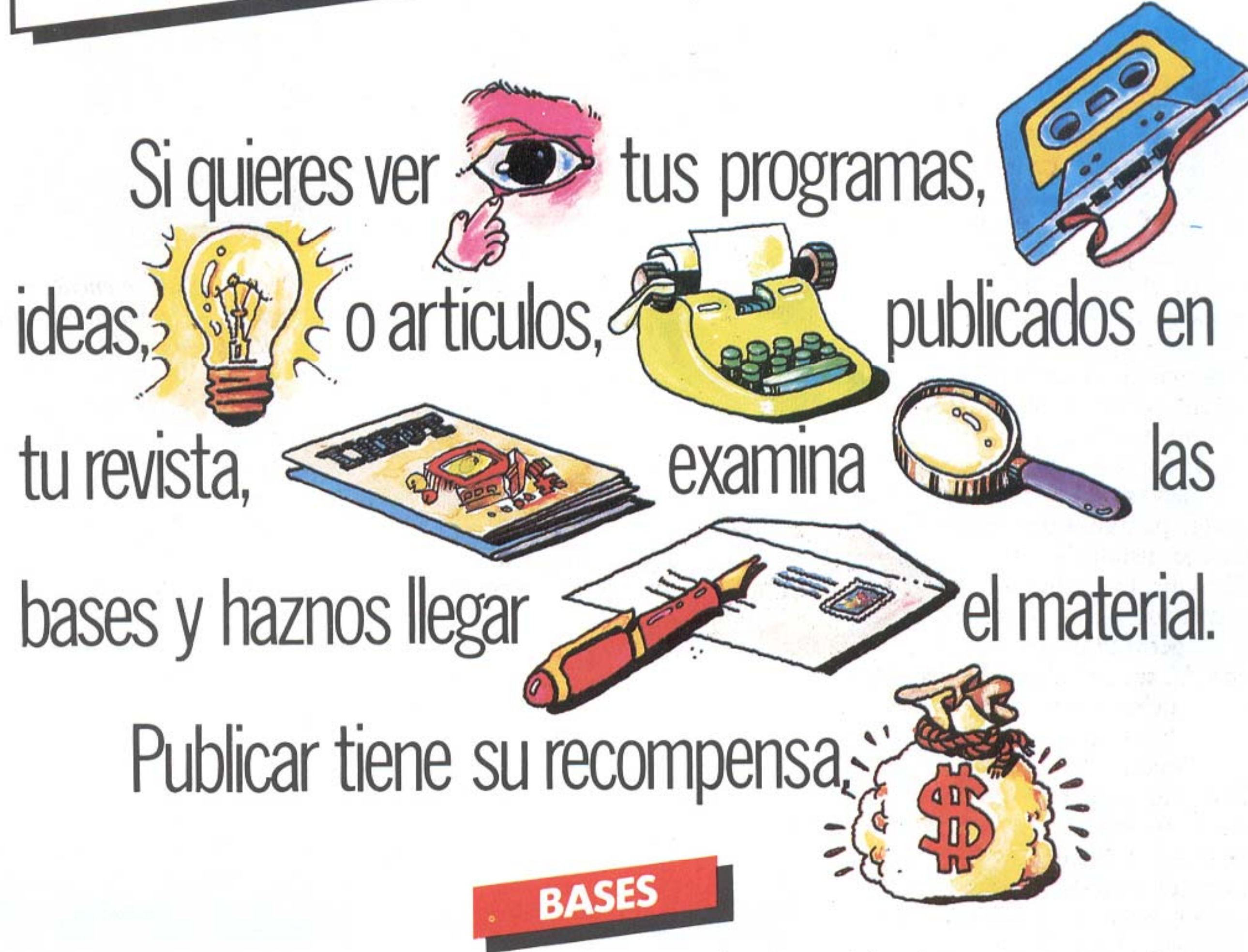
Una de las salas guarda parte del plano secreto que le llevará a la salida.





Nuestro intrépido agente debe tomar decisiones rápidas.

¡PARTICIPA EN INPUT!



BASES

PROGRAMAS: Una vez desarrollado tu programa, que debe ser original y no haber sido enviado a ninguna otra publicación, puedes enviárnoslo aquí grabado en cassette, diskette o microdrive. Es preferible que vaya acompañado por un listado de impresora, pero no es imprescindible.

El programa habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

ARTICULOS E IDEAS: Se aplica lo anteriormente dicho para los textos que acompañan a los programas; es decir, conviene detallar al máximo lo que desees que aparezca publicado en la revista, de la manera que te gustaría que otra persona hubiera explicado eso mismo.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvidéis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante cada mes SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
  en material microinformático a elegir por los afortunados.

¡No os desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas serán revisadas con el máximo interés.

INPUT COMMODORE
Alberto Alcocer, 46, 4.º B
28016 Madrid

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

PANORAMA PARA MATAR

Prácticamente agotados para el *software* los temas de marcianitos, las nuevas tendencias se centran cada vez más en las películas. El juego que examinamos a continuación recoge tres de las secuencias de acción de la película más reciente de **James Bond,** *«A view to a kill»* (Panorama para matar). Se presentó durante el pasado **Commodore Show,** celebrado el mes de junio en Londres, con el atractivo de las chicas **Bond** acudiendo al *stand* de la compañía que lo ha desarrollado. Ahora acaba de llegar a España.
Partiendo del guión original de la película se diseñaron tres juegos, que se traducen en otros tantos módulos perfectamente diferenciados. Probablemente el único punto en común de todos ellos sea la constante de la serie: **James Bond** nunca muere.
Como era de esperar, el juego comienza con una atractiva presentación en la que posa el agente «con licencia para matar» acompañado por su pistola, mientras suenan los acordes de la música.
A diferencia de otros programas, aquí no es el juego quien controla y domina al personaje sino, al revés, es la historia quien gira en torno a él.
Cada uno de los tres juegos se puede afrontar individualmente o se puede avanzar a través de ellos.
El primer módulo cobra vida en París. La malvada **Mayday** acaba de asesinar a un agente en la torre **Eiffel.** Perseguida por **Bond** salta en paracaídas. Tu misión (como **Bond** que eres) consiste en perseguirla por

DATOS GENERALES	CALIFICACION
TITULO: A view to a kill	GRAFICOS: 3 sobre 5
FABRICANTE: Domark	COLOR: 4 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 4,5 sobre 5
MEMORIA: 64 K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Aventura de espías	REALISMO: 4,5 sobre 5



las calles de la capital francesa en un taxi robado, tratando de interceptarla en su punto de aterrizaje. En el laberinto de calles algunas son de dirección única y si tomas varias en dirección prohibida te perseguirá la policía. Si no logras atraparla, el juego se transforma en una persecución de coches por las calles de París.
La pantalla se divide en dos áreas. Arriba una sensacional vista tridimensional, desde la ventana delantera del taxi, evoluciona a media que avanzamos entre los edificios. La otra zona es una vista parcial del plano de la ciudad, en el que se puede apreciar el taxi y el paracaídas, deslizándose de manera que siempre podemos ver la posición que ocupamos. La hora y el día son igualmente visualizados, pues el juego evoluciona en tiempo real. Igualmente, los límites de velocidad vienen impuestos. El control del automóvil no resulta fácil en un principio, pero un poco de práctica lo soluciona.
La segunda parte tiene por escenario el **City Hall** (Ayuntamiento) de San Francisco. Es un juego de aventuras en el que es acompañado por una preciosa chica, **Stacey.** Han sido capturados por el villano **Max Zorin,** que lanza una botella de líquido inflamable en el ascensor donde viajan



Algunas escenas comprometidas de «Panorama para matar». Cortesía de Cinema Internacional Corp.

obligadamente ambos. El indestructible agente logra escapar, pero su rubia acompañante queda atrapada en el edificio en llamas. En nuestra evolución por las diversas salas podemos recoger objetos actuando con el *joystick* sobre los letreros que ofrece la pantalla, como en los juegos de aventuras. Las salas están dibujadas en forma tridimensional y en la parte inferior podemos observar cómo es consumido el edificio por las llamas.

Como es natural, **Bond** logra escapar, porque conoce el camino de salida, pero es el jugador quien debe descubrirlo.

La tercera escena es tal vez la más peligrosa. Bajo el Valle del Silicio existe una mina secreta. **Zorin** ha amenazado con hacer estallar un paquete explosivo y **Stacey** y **Bond** tendrán que desactivar el detonante. En caso contrario, se traducirá en la desaparición de San Francisco y Los Angeles, por la existencia de la conocida falla de San Andrés. Tienes que avanzar por la mina, pero con cuidado, pues las explosiones causan corrientes de agua. **Bond** cae en desgracia y **Stacey** logra escapar. La anteriormente enemiga **Mayday** es rescatada justo a tiempo para ayudar en el salvamento del agente británico. Nuevamente aparecen las opciones, **Bond** puede desactivar el detonador o llevárselo antes de que pase el tiempo límite.

Presentación del juego.



Persecución por París.





El City Hall en llamas.



Entrada real.

CAMPAÑA ANTITERRORISTA

Un grupo de fuerzas terroristas amenaza la seguridad mundial, eres el jefe de un comando antiterrorista y debes hacer todo lo posible para que el diabólico grupo **Cobra** no se adueñe del planeta. Estamos de nuevo ante un juego fuera de serie con unas características muy muy peculiares: vas a visualizar 6 pantallas antes de jugar. En la primera aparecen los miembros de tu grupo alrededor de un tanque. La segunda muestra nuestro planeta con las zonas invadidas por los grupos **Cobra.** Aparecen dos tipos de zona señaladas: una con círculos, otra con estrellas.
Las zonas del globo que aparecen con círculos, indican que la lucha se realizará en campo abierto. Estaremos entonces ante juegos tipo **Zaxxon,** efectos tridimensionales logrados gracias a un *scroll* —desplazamiento horizontal, vertical o diagonal de la pantalla— y una perspectiva: el terreno de batalla se ve desde arriba. Antes de luchar tendrás que elegir a uno de entre cuatro de tus hombres especializados en este tipo de misiones, cada uno de ellos con su respectivo equipamiento: dispones de dos pilotos (uno de cazas y otro de helicópteros), así como dos conductores (tanque y coche blindado). Una vez elegido el hombre, apareces en el lugar de la acción. Con este tipo de pantalla siempre te hallarás ante campos, que según la zona en que te halles serán jungla, desierto o estarán cubiertos por la nieve. El hombre que elijas tendrá que encontrar las fuerzas de **Cobra** y como es lógico destruirlas.
Si logra su empeño verás de nuevo el mapa del mundo en el que tendrás que elegir la próxima misión. Si esta vez optas por mandar a uno de tus hombres a una zona indicada con una estrella, cambia el tipo de juego. De buenas a primeras volverás a ver la pantalla de tus oponentes, aunque ya con una ligera variación respecto a la original: el enemigo contra el que luchaste aparecerá entre rejas (en caso claro está de que lo hayas vencido, si no es así aparecerá tu combatiente en esta desagradable situación).



La lucha va desde el enfrentamiento hombre a hombre hasta el empleo de armamento pesado.



Podemos elegir a nuestro hombres y armas para el ataque.





Los puntos calientes del globo.

DATOS GENERALES	CALIFICACION
TITULO: Gi-Joe	GRAFICOS: 5 sobre 5
FABRICANTE: Epix	COLOR: 4,5 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 4,5 sobre 5
MEMORIA: 64 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Aventura antiterrorista	REALISMO: 3,5 sobre 5

Tendrás que elegir a hombres de infantería (para las commodoreras feministas hay que decir que también hay mujeres), cada uno está especializado en un terreno concreto y, por tanto, tiene armas especiales —lo que no impide que elijas al hombre de las nieves para

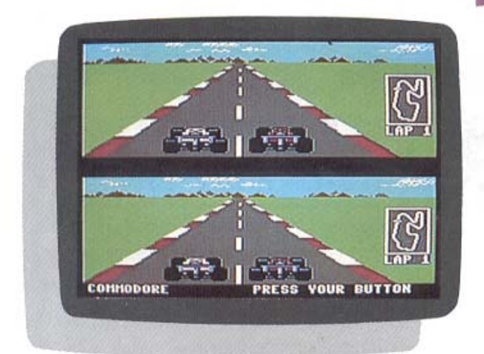


En el cuartel general permanece acuartelado el ejército antiterrorista más preparado de los EE. UU. (por supuesto).

luchar en la jungla—. Tu hombre luchará en una pantalla estática contra el miembro del comando **Cobra.** Esta pantalla casi siempre será diferente, y la definición es la normal de esta casa... hemos de admitir que nos tiene muy mal acostumbrados.
En fin, el juego no tiene pérdida (sólo la de tus hombres). En cuanto a lo que al juego se refiere, las pantallas con *scroll* requieren cierta práctica en el manejo del *joystick*, ya que los **Cobra** no tienen «un pelo de ineptos», en las pantallas estáticas tampoco hay que ser demasiado despistado, porque tu oponente suele tener un robot que le ayuda en su tarea de aniquilación.
Como cabe esperar la conclusión es que éste es un juego que te garantizará horas y horas de entretenimiento.

UN DIA EN LAS CARRERAS

Conducir un coche de fórmula es una ilusión que no parece estar al alcance de cualquiera... que no tenga un **C-64.** El programa que pasamos a describir es el complemento que convierte a tu cuarto en el **Jarama** o **Le Mans.**
El **Pitstop II** sigue la línea marcada por el mítico **Pole Position,** que crease **Atari** para las máquinas de los salones recreativos y más tarde para sus ordenadores.
Aquí la primordial innovación que se aporta es la disposición de una pantalla para cada uno de los dos principales contendientes, aunque también está previsto que solamente sea uno.
A este programa le han sido añadidas notas de realismo, tales como las detenciones por causas técnicas para repostar combustible y cambiar los neumáticos, que al comienzo de la carrera son negros, pero su color va aclarando a medida que aumenta su



La carrera está a punto de dar comienzo. Sólo falta presionar el botón de disparo del joystick.

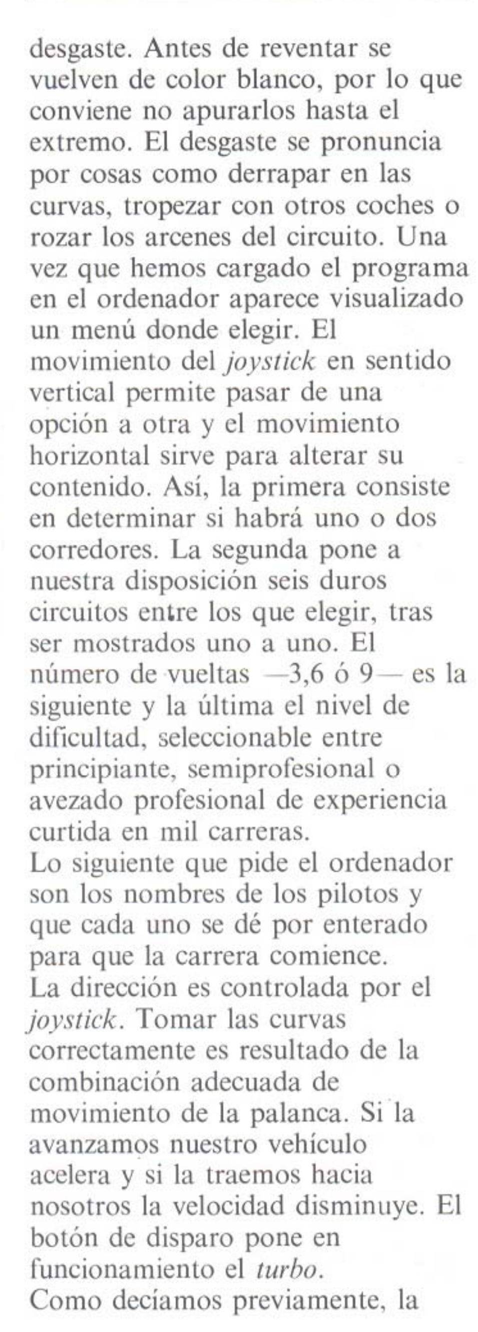


Sólo el más preparado ganará. Es importante llevar un control estricto de los tiempos y el consumo.



Durante la carrera podemos repostar y cambiar los neumáticos que hayan sufrido mayor desgaste.

DATOS GENERALES	CALIFICACIONES
TITULO: Pitstop II	GRAFICOS: 5 sobre 5
FABRICANTE: Epyx	COLOR: 4 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 4,5 sobre 5
MEMORIA: 64 K	INTERES: 4,4 sobre 5
CLASE DE PROGRAMA: Composición automovilística	REALISMO: 4,5 sobre 5



desgaste. Antes de reventar se vuelven de color blanco, por lo que conviene no apurarlos hasta el extremo. El desgaste se pronuncia por cosas como derrapar en las curvas, tropezar con otros coches o rozar los arcenes del circuito. Una vez que hemos cargado el programa en el ordenador aparece visualizado un menú donde elegir. El movimiento del *joystick* en sentido vertical permite pasar de una opción a otra y el movimiento horizontal sirve para alterar su contenido. Así, la primera consiste en determinar si habrá uno o dos corredores. La segunda pone a nuestra disposición seis duros circuitos entre los que elegir, tras ser mostrados uno a uno. El número de vueltas —3,6 ó 9— es la siguiente y la última el nivel de dificultad, seleccionable entre principiante, semiprofesional o avezado profesional de experiencia curtida en mil carreras.

Lo siguiente que pide el ordenador son los nombres de los pilotos y que cada uno se dé por enterado para que la carrera comience.

La dirección es controlada por el *joystick*. Tomar las curvas correctamente es resultado de la combinación adecuada de movimiento de la palanca. Si la avanzamos nuestro vehículo acelera y si la traemos hacia nosotros la velocidad disminuye. El botón de disparo pone en funcionamiento el *turbo*.

Como decíamos previamente, la pantalla queda dividida en dos, horizontalmente. La mitad superior sirve para el primer corredor y la inferior, idéntica, para el segundo. Esta necesidad proviene del hecho de que cada participante no tiene por qué estar en el trozo de cirucuito visualizado, pues la vista es parcial y corresponde a una perspectiva realista, en la que el fondo lejano se desplaza en la medida en que el automóvil gira (y por tanto nuestro punto de vista).

Debajo de cada mitad aparece la escueta información de interés para el piloto: la velocidad, el tiempo y la cantidad de gasolina que hay en el depósito. Incorporado en cada vista aparece un mapa del cirucuito y un punto brillante indica dónde nos encontramos en cada momento.

Cuando precisemos asistencia podemos ir a nuestra cochera, para que el competente equipo técnico haga los ajustes de rigor. Cuando nos aproximamos a la entrada, situada en la parte izquierda de la pista, habrá que girar la dirección *(joystick)* en esa dirección para encaminarlo. Allí nos esperan dos experimentados mecánicos. Podemos elegir con el *joystick* cuál de ellos entra en acción.

Supongamos que ha llegado el instante de repostar. Movemos la cruz que aparece en nuestra mitad de la pantalla directamente con el *joystick;* la presión del botón de disparo le hace entender que nos dirigimos a él y con un ligero movimiento le acercamos al coche para dar comienzo a su tarea. A medida que el indicador sube, oímos sonar las clásicas campanillas del surtidor de gasolina. El otro asistente puede cambiar mientras tanto cualquier neumático (o todos), controlando igualmente por el *joystick*. El efecto de verlos trabajando tiene un realismo sorprendente. La vuelta al circuito se logra con sólo apuntar al corredor con la cruz y presionando el consabido botón, para que vuelva a rugir el motor.

Revista de Software

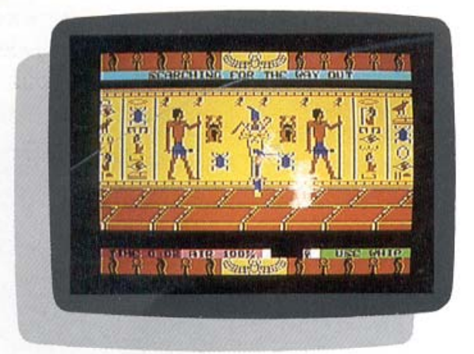
TUTANKAMON LEVANTA LA CABEZA

El noble británico *sir* **Arthur Pendragon** se ve envuelto en una fantástica aventura, conducido por su verdadera y única vocación: la Arqueología. Su sed de conocimientos le introduce en un laberinto faraónico que le puede traer serios disgustos. Haciendo indagaciones en una antigua tumba logra dar con la clave que le permite entrar, pero... queda atrapado por la gracia del **Ojo de Osiris.**

Tras acostumbrar sus ojos a la tenue luz producida por las antorchas, *sir* **Arthur** se da cuenta de cuál es su entorno, una sala mortuoria presidida por dos gigantescas estatuas de piedra en postura sedente. Estas inmensas moles guardan la entrada y son capaces de juntarse o separarse deslizando, impidiéndole el acceso a la salida siempre que intenta acercarse. El aire es cada vez más escaso. Hay que encontrar la clave que deja el camino expedito. Una vez hallada, *«era fácil. Se trataba únicamente de saltar en el rincón adecuado»*. El movimiento del *joystick* permite al *sir* moverse en tantas direcciones como dispone esta útil palanca. Una



Para poder salir, sir Arthur debe saltar en un determinado rincón. Las estatuas le abrirán el camino.



En los corredores es normal toparse con la momia, en un clásico ambiente protagonizado por jeroglíficos.

presión en el botón de disparo le permite dar elevados brincos.

Camina despistado por los corredores y pasillos de la tumba, llenos de jeroglíficos. Es posible acceder a

multitud de salas, pero el camino no es fácil, multitud de mosquitos, extrañas abejas y pájaros no cesan de molestarle. Una momia y un escorpión gigantescos se suman a la fiesta. En una sala guardada por enormes bolas rodantes, que le impiden el avance, encuentra —en el interior de un cofre— un látigo como el empleado por **Indiana Jones.**

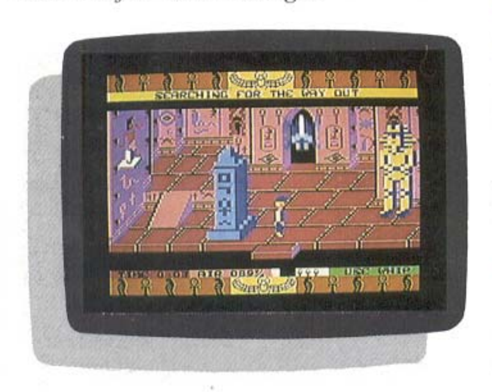
Sir **Arthur** puede realizar tres tipos de acción, lo cual acerca a este juego a los clásicos de aventuras en los que se puede elegir la acción que ha de seguir. Estos son: saltar, dar latigazos o emplear la antorcha. Una simple presión sobre cualquier tecla permite hacer la elección, que aparece reflejada en una etiqueta existente al pie de la pantalla. Lamentablemente no es posible

utilidad para las salas no iluminadas.
La salida definitiva podrá ser localizada cuando logre descifrar los pergaminos que le proporcionan las pistas suficientes.
Una interesante facilidad que da el juego son las pausas. En efecto, presionando la tecla F7 podemos detener la marcha del programa y trazar el mapa que nos ayudará a



En el cofre está el látigo.

DATOS GENERALES	CALIFICACION
TITULO: Emtombed	GRAFICOS: 5 sobre 5
FABRICANTE: Ultimate (producido por abc soft)	COLOR: 4 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 4,5 sobre 5
MEMORIA: 64 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Aventura arqueológica	REALISMO: 3,5 sobre 5



hacerlo todo a la vez, por lo que se debe optar cuidadosa y rápidamente por la acción más adecuada al momento. Queda por aclarar que *sir* **Arthur** debe localizar previamente el látigo y la antorcha para poder utilizarlos posteriormente. El primer elemento sirve para mantener alejadas a las criaturas que le asedian sin parar, el segundo se revela de gran

que *sir* **Arthur** no cometa siempre los errores, que le hagan terminar en corredores y salas sin continuación. Si en un momento dado decidimos volver a comenzar desde el principio, la tecla F1 es la solución. El número de pantallas es ilimitado y continuamente aparecen nuevas sorpresas. Salir con bien parece una tarea imposible, pero es factible.

****

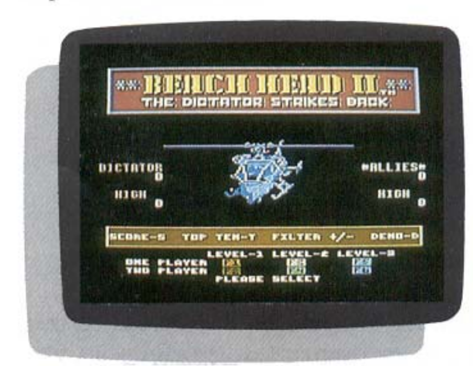
¿ERES EL BUENO O EL MALO?

Tenemos ante nosotros un juego de guerra, aunque no tiene nada que ver con la estrategia, los movimientos de tropas, etc. Unicamente se trata de cruel y encarnizado combate.
El subtitulo reza: «El Dictador ataca de nuevo», o, lo que es igual, buenos contra malos.
Repasemos la trama. El Dictador ha capturado como rehenes a las

tropas enemigas. Lograr su rescate es el desafío.
En la pantalla de presentación aparece el helicóptero que nos introduce en la primera parte. Bajo él un breve menú en el que elegir uno de entre tres niveles de dificultad. Igualmente puede optar

Menú de presentación para elegir.

por uno o dos jugadores; es decir, que puedes perfectamente ser tú el Dictador que intenta la derrota de las tropas aliadas, encabezadas por tu padre o hermano.



Para intentar el rescate de las tropas recluidas se lanzarán las fuerzas paracaidistas desde el helicóptero, durante esa primera fase del juego.
Un enorme y destructivo cañón amenaza a los paracaidistas, escondidos tras un muro después de su toma de tierra. Una brecha en el primer muro te permite llegar a otro más próximo al cañón, primer objetivo que debe ser alcanzado para entrar a la fortaleza. El peligro de salir a campo abierto es el consabido de estar a merced de los proyectiles de la artillería del Dictador, los cuales habrá que burlar. Los supervivientes, si los hay, habrán de tomar finalmente el cañón y neutralizarlo. Por tanto la misión de estas fuerzas de choque es fundamental. La estrategia empleada consiste en que los hombres salen por parejas, el primero procura atraer el fuego sobre sí, mientras el otro lanza una granada y sale corriendo hacia un lateral para esquivar los disparos. Cubierto el primer objetivo, llegamos a la segundo pantalla en la que el resto de la tropa debe atravesar el patio de la fortaleza. Con el cañón recién tomado al enemigo, cubrirás el ataque de tus hombres. Es imprescindible aniquilar a los cuatro adversarios que oponen toda su resistencia contra el avance. El más peligroso es uno que se esconde tras una trampilla móvil por todo su recorrido. Se desplaza con enorme rapidez y continuamente sale un enemigo que dispone bombas a dos pasos de tus hombres. Por ser el más conflictivo es el primer adversario que debes eliminar. Siguiéndole en peligrosidad está un automóvil que te persigue implacablemente y una vez alcanzado te desintegra sin piedad. Los dos últimos obstáculos son un tanque, que intenta aplastarte mediante el atropello, y un soldado escondido tras las murallas que dan al patio, por donde lanza mortíferas bombas.
Si logras llegar a la tercera fase, debes elegir una ruta de escape. Pilotas un helicóptero, te persiguen misiles, te atacan desde *bunkers,* desde trampas escondidas en el suelo lanzan pelotas. Un sinfín de peligros te obstaculizarán la entrada a la madriguera del Dictador. La cuarta etapa es un enfrentamiento personal entre los dos líderes: el comandante **Striker** y el Dictador. En un gruta les separa un precipicio por el que avanza el curso de un río. Entonces hablan los cuchillos, que se lanzan mutuamente ambos contendientes con la intención de dar muerte al otro. El vencido cae y es arrastrado por las frías aguas del río, no sin antes dar un espantoso grito inhumano, que guarda gran semejanza con el que produce el agente de otro juego, **Misión imposible,** cuando cae al abismo.
No hemos mencionado aún que este juego dispone de una serie de gritos que lanzan los hombres, un conjunto de unas treinta palabras y frases distribuidad por todo él; por ejemplo, cuando un soldado cae abatido por un proyectil grita: «Me han disparado», eso sí, en inglés.
Aunque el juego no aporta nada nuevo del otro mundo en cuanto a los gráficos, la emoción está asegurada.

DATOS GENERALES	CALIFICACION
TITULO: Beach Head II	GRAFICOS: 3 sobre 5
FABRICANTE:	COLOR: 3,5 sobre 5
ORDENADOR: Commodore 64	PRESENTACION: 3 sobre 5
MEMORIA: 64 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Juego de guerra	REALISMO: 3,5 sobre 5



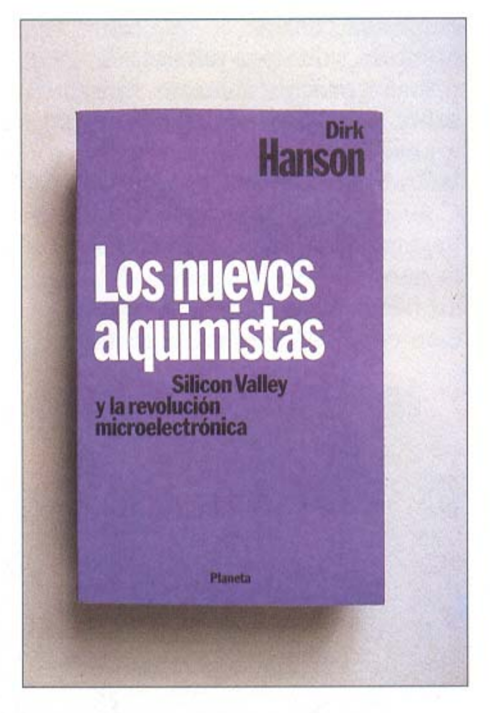
El acceso al patio no es sencillo.



Los paracaidistas han de tomar el cañón gigante, situado en la parte inferior de la pantalla.



La lucha final a muerte entre ambos líderes terminará con la victoria del cuchillo de uno de ellos.



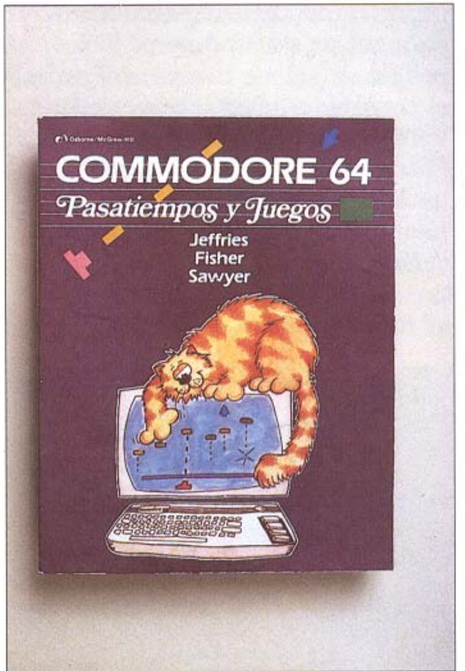
LOS NUEVOS ALQUIMISTAS

Autor: Dirk Hanson
Editor: Planeta
Páginas: 292
Precio: 800

Los nuevos alquimistas es una narración amena y detallada de la evolución de la tecnología electrónica desde los tiempos en que Edison fabricó su lámpara de incandescencia hasta nuestros días, repasando con especial detenimiento las circunstancias que propiciaron la aparición y el desarrollo, en Silicon Valley (Valle del Silicio), California, de la industria de los semiconductores. Es la historia de los hombres que consiguieron transmutar el silicio en oro o, lo que es lo mismo, la tecnología en un negocio en el que se invierten y se ganan —o se pierden— millones y millones de dólares.

Dirk Hanson relata el rápido progreso de la electrónica describiendo sus hitos: los primeros ordenadores, el transistor, los circuitos integrados, los microprocesadores, ordenadores personales, etc. Pero también cuenta las circunstancias que impulsaron y dieron forma a este desarrollo. Así, la importante influencia de los militares, que aportaron tanto su avidez de tecnología como sus tentadores presupuestos.

Analiza, además, las implicaciones y consecuencias que el vertiginoso progreso electrónico está teniendo en nuestra sociedad, e intenta hacer una prospección de lo que será en el futuro: la sociedad automatizada.



COMMODORE 64 PASATIEMPOS Y JUEGOS

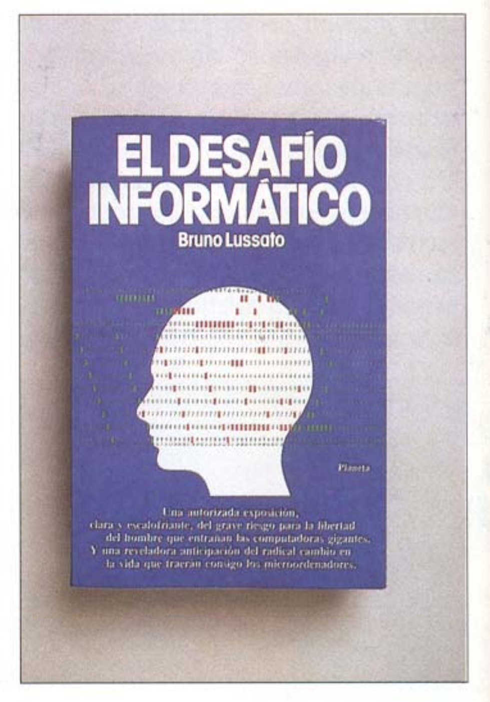
Autor: Jeffries, Fisher y Sawyer
Editor: Osborne/McGraw-Hill
Páginas: 182
Precio: N.D.

Caja fuerte, Bonzo, Ajedrez indio, Rescate, Captura, Inmersión, etc., y así hasta un total de 35 títulos presentan a otros tantos programas en un libro que, como su portada adelanta, va destinado al ordenador **Commodore 64.**

En las primeras páginas se incluyen cinco programas cortos que llevan la misión implícita de iniciar a los novicios. Igualmente sirven para familiarizar al lector con la notación especial utilizada en todos los listados de la obra. Los espacios en blanco son indicados con una pequeña flecha, los caracteres gráficos producidos mediante la combinación de teclas no aparecen en la forma habitual, sino que se detallan todas ellas por el orden en que deben ser pulsadas. No obstante, la introducción ofrece detenidamente todos los detalles necesarios.

El original corresponde a una obra de producción estadounidense, magníficamente adaptada a nuestro idioma por **Luis Joyanes,** quien goza de una larga trayectoria en la creación y adaptación de títulos particularmente orientados hacia **Commodore 64.**

Conviene resaltar que los listados han sido elaborados con tipografía de imprenta, huyendo de las clásicas reproducciones fotográficas de listados procedentes de una impresora matricial.



EL DESAFIO INFORMATICO

Autor: Bruno Lussato
Editor: Planeta
Páginas: 202
Precio: 700

El ordenador se plantea como un reto que todos deberemos afrontar en el futuro. Sin embargo, los agoreros se anticipan a presentarlo con una gran dosis de incertidumbre para el pobre mortal. A su entender los sistemas informáticos comienzan a significar una seria amenaza.

Bruno Lussato, el autor de este libro, parte de un supuesto diametralmente opuesto, y su intención es arrojar luz sobre el tema de la forma más coloquial posible.

El miedo desaparece con la información, parece ser la moraleja escondida tras **El desafío informático.** A lo largo de las páginas condimenta las peculiaridades y evolución de la industria informática con alusiones a la literatura y otras artes más conocidas por todos.

Esta obra desprende, casi de inmediato, la enorme fe que su autor tiene en el microordenador, como algo necesario para hacer que evolucione la nueva sociedad.

Fundamentalmente estamos ante una obra destinada a quienes desean conocer más a fondo los recursos ofrecidos por el ordenador. No se puede adjudicar a su contenido el calificativo de técnico, pero sí divulgativo.

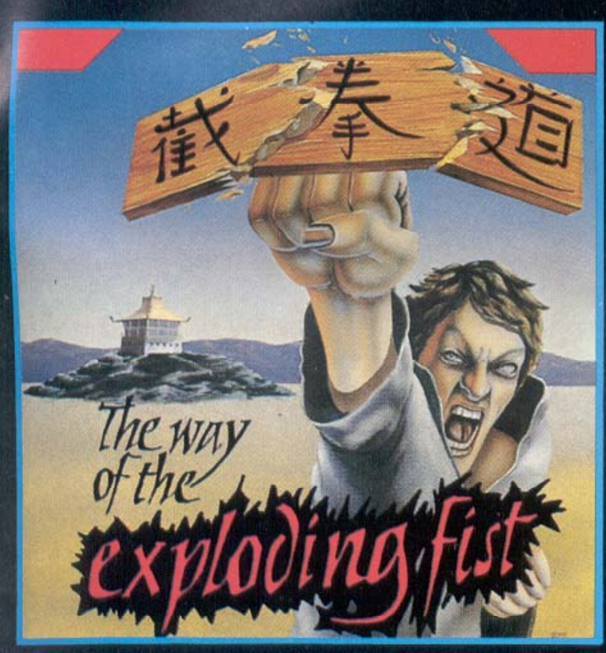


Tiene lo mejor para tu COMMODORE

EXPLODING FIST

El primer juego de kàrate para el C-64. Cuidadosamente diseñado para simular todos los movimientos de las tradicionales artes marciales. Increibles gráficos y música oriental hacen que este programa sea ya nominado como el "Juego del año" por la mayoría de las críticas.

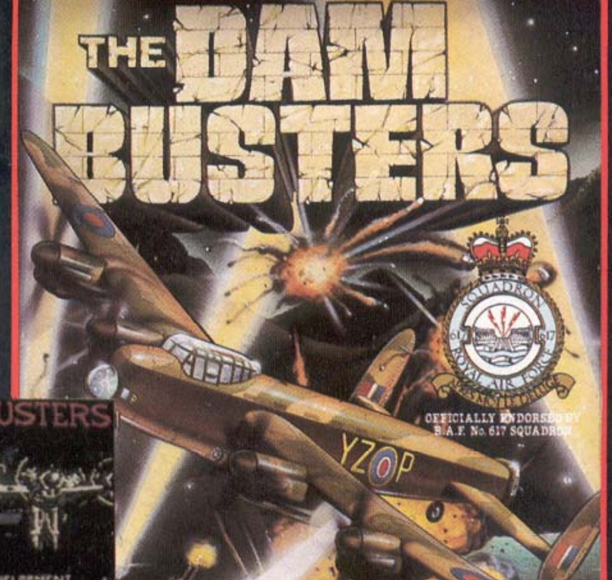






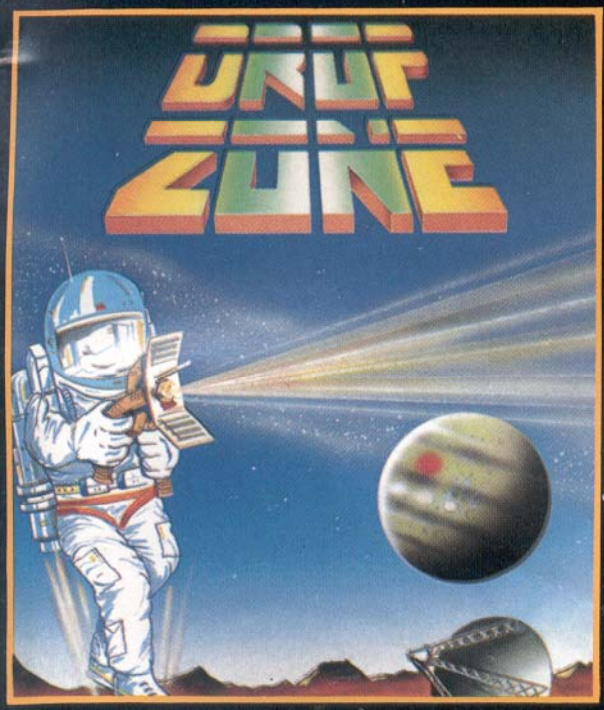
DAMBUSTERS

Reproducción completa de la misión que en la II Guerra Mundial llevaron a cabo los ingleses para destruir las más importantes presas que abastecían de energía a las fábricas de armamento alemanas. La mezcla de simulador de vuelo, estrategia y arcade hacen de DAMBUSTERS, el programa más vendido en U. K. para COMMODORE.



DROP ZONE

Sencillamente un sueño hecho realidad. Por fin un juego que mantendrá a los aficionados a los programas de acción, entretenidos durante meses. Rapidez de juego y total control, unidos a efectos sonoros y gráficos fabulosos hacen de este juego de corte espacial, el primero de su categoría. ¡No te lo pierdas!

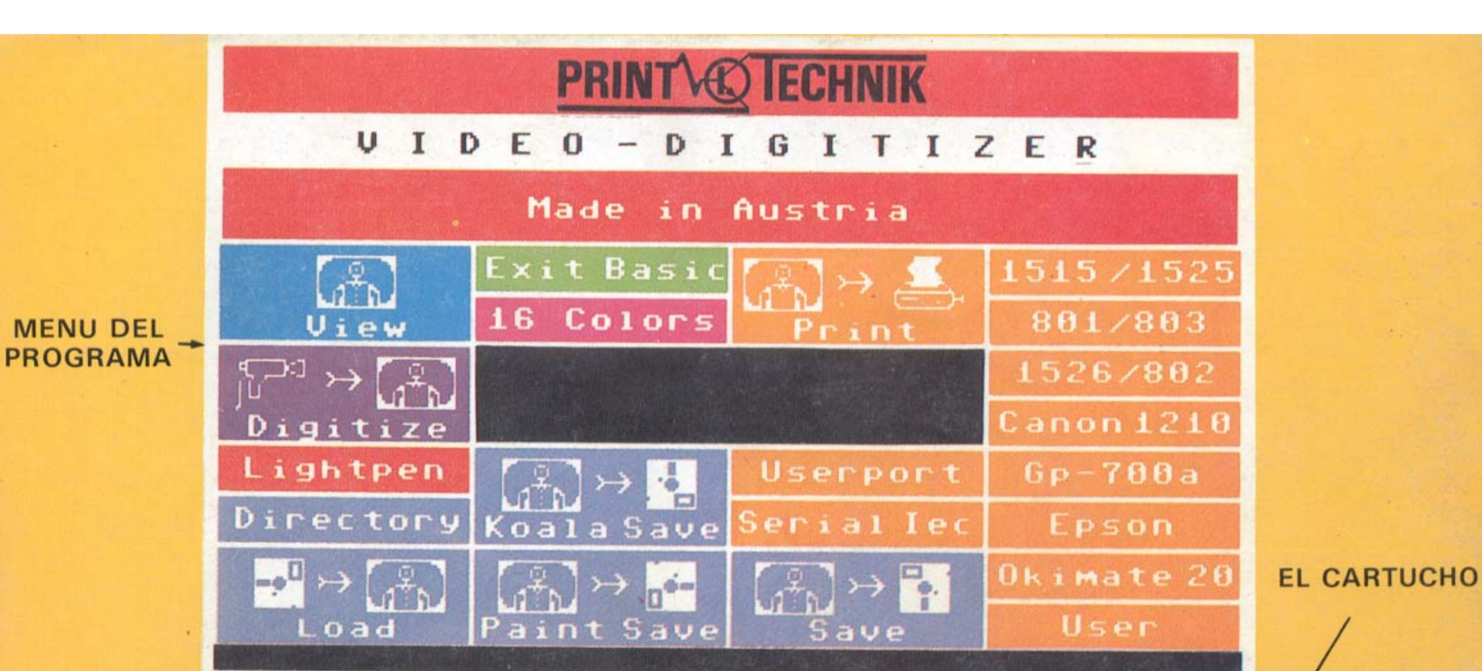


GREMLINS

Siéntete protagonista de la película con esta maravillosa aventura íntegramente en castellano. Más de 100 pantallas diferentes en las que tendrás que poner a prueba tu destreza e inteligencia.



Si deseas recibir información completa de nuestros programas para el COMMODORE 64, escribenos a ERBE, c/. Santa Engracia, 17, 28010 Madrid ó llámanos al (91) 447 34 10. Te haremos llegar nuestro catálogo de forma gratuita.



MENU DEL PROGRAMA

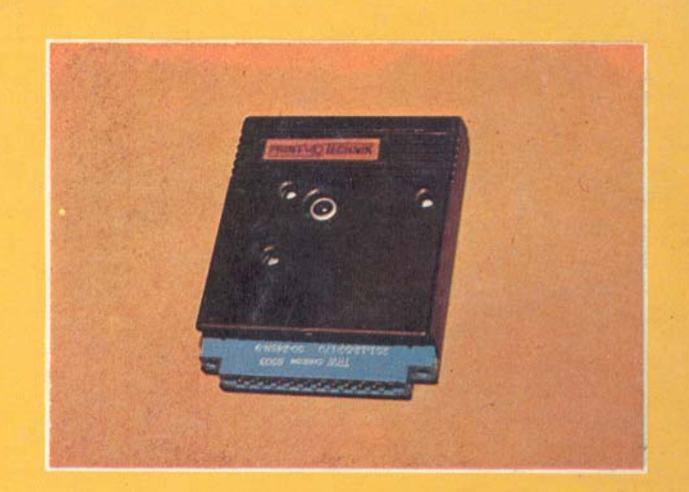
EL VIDEO - DIGITALIZADOR de imagen es un módulo que le permite digitalizar señales (imágenes) de video, almacenándolas en la memoria del ordenador en el formato de gráficos de alta resolución.

Las imágenes digitalizadas, pueden ser visualizadas, almacenadas en disco según diferentes formatos, modificadas con ayuda de un Lapiz Optico o Koala o impresas en una gran gama de impresoras.

EL VIDEO - DIGITALIZADOR de imagen es un pequeño cartucho que se conecta al port de usuario y esta provisto de un conector que le une al emisor de video. Este puede ser: una cámara de video, un receptor de TV con salida de video, o cualquier sistema grabador-reproductor de video.

Las aplicaciones son ilimitadas y sólo su imaginación le pueden poner fronteras.

EL CARTUCHO

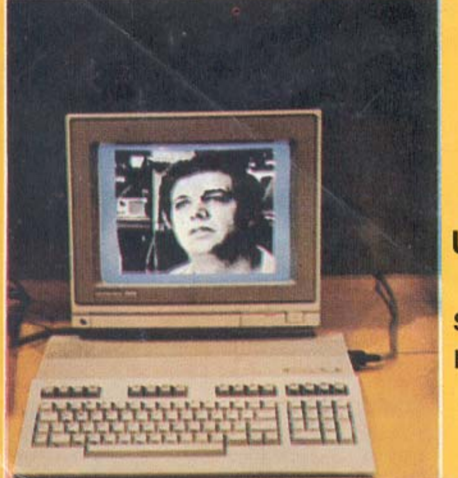




VIDEO DIGITALIZADOR EN ACCION



SALIDA POR IMPRESORA MPS 801



UN NUEVO Y SOFISTICADO MUNDO SE ABRE A SU ORDENADOR

SOLICITELO A SU DISTRIBUIDOR HABITUAL O A:



Santa Cruz de Marcenado, 31 - 28015 Madrid - Teléf. 241 10 63